

UNIVERSIDADE DE LISBOA

Faculdade de Ciências

Departamento de Informática



**AVATARES EM LÍNGUA GESTUAL
PORTUGUESA**

José Carlos Malhado Bento

PROJETO

MESTRADO EM INFORMÁTICA

2013

UNIVERSIDADE DE LISBOA

Faculdade de Ciências

Departamento de Informática



**AVATARES EM LÍNGUA GESTUAL
PORTUGUESA**

José Carlos Malhado Bento

PROJETO

MESTRADO EM INFORMÁTICA

Trabalho orientado pela Professora Doutora Ana Paula Boler Cláudio e co-orientado pelo
Professor Doutor Paulo Jorge Cunha Vaz Dias Urbano

2013

Agradecimentos

Para mim não seria possível ter concretizado esta tese sem um conjunto de pessoas que me rodearam durante esta realização pessoal e académica. Agradeço em primeiro lugar aos meus pais pelas condições que me criaram para desenvolver este trabalho, especialmente à minha mãe pela sua preciosa ajuda com o seu contributo e conselhos.

O meu especial obrigado vai para os meus orientadores, a Doutora Ana Paula Cláudio e Doutor Paulo Urbano que me apoiaram com a sua orientação e tive a sua compreensão durante este percurso. Tenho a certeza que produzi um documento muito mais científico do que aquele que existiria sem a sua participação.

Um obrigado também especial ao José Maria Dinis, porque no âmbito desta tese, foram muito importantes os seus ensinamentos e explicações. Também quero agradecer além do grande amigo que é o Marcos, também contribuiu para esta tese, juntamente com o “*Guns*”, o Tiago Antunes, o “*Faísca*” e o Rafael que deram-me uma ajuda fundamental no meu percurso curricular.

Por último, não menos importante, agradeço à minha namorada pelo apoio e muita paciência que teve comigo.

Aos meus avós

Resumo

Em Portugal existem 33 mil surdos que usam a Língua Gestual Portuguesa como sua língua materna. Esta língua tem um léxico, regras sintáticas e semânticas, tal como uma língua oral. Os intérpretes qualificados da LGP escasseiam, pelo que faz sentido recorrer a avatares capazes de reproduzir esta língua. As principais vantagens no uso de avatares são que a animação de um gesto pode ser facilmente manipulada e reutilizada. Estas propriedades tornam possível a combinação exponencial de gestos de palavras em frases de língua gestual, ao contrário do vídeo que teria de ser refeito totalmente sempre que se entenda alterar ou adicionar um gesto.

Recentemente têm proliferado as tecnologias de baixo custo, com especial destaque para o sensor Microsoft Kinect[®], que permitem a aquisição de dados do movimento humano. Embora os dados recolhidos pelo sensor sejam limitados ao nível dos dedos, tornam o processo de captura de movimentos muito mais acessível e prático do que os sistemas convencionais.

Neste sentido, explorei um conjunto de técnicas para a aquisição do movimento humano e tradução do mesmo para um personagem virtual. Como resultado dessa exploração, adotei uma abordagem híbrida que utiliza o sensor Kinect[®] para gravar o movimento corporal e facial, enquanto o movimento dos dedos é animado manualmente numa ferramenta de animação.

Foi desenvolvida uma prova de conceito que usa um avatar como intérprete gestual que de uma forma divertida introduz conceitos básicos de LGP dirigido a surdos e ouvintes, e em particular a crianças. Os resultados da avaliação foram muito interessantes e mostraram que é útil uma aplicação deste género.

Palavras-chave: língua gestual portuguesa, LGP, avatar, animação, captura de movimentos, crianças, ensino, kinect.

Abstract

In Portugal there are 33,000 deaf people that use Portuguese Sign Language (PSL) as their mother language. This language has a lexicon, a syntactic and semantic rules such as an oral language. The qualified interpreters of PSL are scarce, so it makes sense to use avatars capable of reproducing this language. The main advantages in the use of avatars are the ease of manipulation and reusability of their animation. These properties make it possible to exponential combine sign gestures to create sentences, unlike the video which has to be totally remade whenever it is necessary to change or add a sign.

Recently the dissemination of low-cost technologies, with special attention to the Microsoft Kinect sensor, which allows the acquisition of motion data. Although the collected data is not accurate on finger level precision, makes the process of motion capture more accessible and practical than the conventional systems.

In this sense, we explored a set of techniques to translate human movement into a 3D character. As a result of this development, I adopted a hybrid approach that uses the Kinect sensor to record the body and facial movement, while the finger movement is animated with a manual method by a software animation tool.

It was developed a proof of concept that uses an avatar as an sign language interpreter that in a enjoyable environment, introduces basic concepts of SPL designed to deaf and hearing people, and in particular to children. The results of the evaluation were very interesting and showed that it is useful an application of this genre.

Keywords: portuguese sign language, PSL, avatar, animation, motion capture, children, education, kinect.

Conteúdo

Lista de Figuras	19
Lista de Tabelas	23
Listagens.....	25
Capítulo 1 Introdução	26
1.1 Motivação.....	26
1.2 Objetivos.....	27
1.3 Contribuições / Trabalho realizado	28
1.4 Metodologia de desenvolvimento	29
1.5 Estrutura do documento.....	29
Capítulo 2 Conceitos e Trabalho Relacionado	31
2.1 Conceitos.....	31
2.1.1 Língua Gestual Portuguesa	31
2.1.2 Avatar	33
2.1.3 Captura de movimentos	36
2.1.4 Sensores de profundidade.....	37
2.2 Trabalhos Relacionados.....	41
2.2.1 CopyCat.....	41
2.2.2 Ensino da Língua Gestual Assistido por Personagens 3D Virtuais.....	42
2.2.3 Simon-the-Signer	44
2.2.4 TESSA	45
2.2.5 Visia.....	45
2.2.6 eSign	46
2.2.7 Curso de Língua Gestual em DVD	48
2.3 Dicionários Gestuais	48
2.3.1 Spread the Sign.....	48
2.3.2 Signing Savvy	49
2.3.3 Dicionários Gestuais da VCom3D	50
2.3.4 Análise comparativa entre Dicionários Gestuais.....	51
2.4 Sumário.....	51

Capítulo 3	Investigação e Testes Prévios.....	52
3.1	<i>Escolha do software de animação 3D</i>	<i>52</i>
3.2	<i>Adaptação e primeiros testes em animação 3D.....</i>	<i>53</i>
3.3	<i>Soletrar palavras em LGP – criação de plugin</i>	<i>54</i>
3.4	<i>Animação de gestos assistida por Kinect</i>	<i>55</i>
3.5	<i>Definição de gesto por coordenadas – criação de plugin.....</i>	<i>59</i>
3.6	<i>Hand tracking – desenvolvimento de algoritmos.....</i>	<i>61</i>
3.7	<i>Finger tracking - testes</i>	<i>63</i>
3.8	<i>Finger tracking semiautomático – aplicação Pose Estimation.....</i>	<i>66</i>
3.9	<i>Sumário.....</i>	<i>73</i>
Capítulo 4	Geração e animação do Avatar	74
4.1	<i>Protótipo base - primeira iteração</i>	<i>74</i>
4.1.1	<i>Visualizar frases em LGP</i>	<i>75</i>
4.1.2	<i>Avaliação com surdos na Associação Portuguesa de Surdos</i>	<i>77</i>
4.2	<i>Novo avatar - segunda iteração</i>	<i>79</i>
4.2.1	<i>Concepção de novo Avatar.....</i>	<i>80</i>
4.3	<i>Novos gestos, captura facial e aplicação - terceira iteração.....</i>	<i>82</i>
4.3.1	<i>Produção de novos gestos.....</i>	<i>82</i>
4.3.2	<i>Captura facial.....</i>	<i>83</i>
4.3.3	<i>Aplicação.....</i>	<i>85</i>
4.4	<i>Sumário.....</i>	<i>85</i>
Capítulo 5	Os meus primeiros gestos.....	86
5.1	<i>Motivação.....</i>	<i>86</i>
5.2	<i>Os meus primeiros gestos.....</i>	<i>86</i>
5.3	<i>Público-alvo</i>	<i>89</i>
5.4	<i>Engenharia de Requisitos</i>	<i>89</i>
5.4.1	<i>Levantamento de Requisitos</i>	<i>89</i>
5.4.2	<i>Análise de Requisitos</i>	<i>90</i>
5.5	<i>Desenho do Sistema</i>	<i>93</i>
5.5.1	<i>Arquitetura da aplicação</i>	<i>94</i>
5.5.2	<i>Estrutura de Dados</i>	<i>95</i>

5.5.3	Diagrama de classes	96
5.5.4	<i>Design</i> e Conceção de <i>Interfaces</i>	97
5.6	<i>Implementação</i>	104
5.6.1	Bibliotecas / <i>API's</i>	105
5.6.2	Sistema de Autenticação	105
5.6.3	Front-end	107
5.6.4	Backoffice	114
5.6.5	Problemas encontrados.....	114
5.7	<i>Resultados</i>	116
5.8	<i>Sumário</i>	116
Capítulo 6	<i>Avaliação de “Os meus primeiros gestos”</i>	117
6.1	<i>Objetivo</i>	117
6.2	<i>Participantes</i>	117
6.3	<i>Métricas</i>	117
6.4	<i>Método</i>	119
6.5	<i>Resultados</i>	119
6.5.1	Primeiro questionário – teste ao Dicionário	120
6.5.2	Segundo questionário – teste ao modo Adivinhar.....	121
6.5.3	Terceiro questionário – Perguntas genéricas.....	122
6.6	<i>Sumário</i>	123
Capítulo 7	<i>Conclusão</i>	124
7.1	<i>Dificuldades encontradas</i>	126
7.2	<i>Opinião Crítica</i>	127
7.3	<i>Trabalho futuro</i>	127
Bibliografia	130
ANEXOS	135
	<i>Anexo 1 – Contexto sociocultural dos surdos em Portugal</i>	136
	<i>Anexo 2 – Especificações técnicas e algoritmo usado pelo Kinect</i>	137
	<i>Anexo 3 – Questionário da avaliação do protótipo inicial para componente do Soletrar palavras em LGP</i>	138

<i>Anexo 4 – Questionário da avaliação do protótipo inicial para a visualização de frases em LGP.....</i>	<i>139</i>
<i>Anexo 5 – Descrição formal do caso de uso Traduzir e Adivinhar gesto de palavras .</i>	<i>141</i>
<i>Anexo 6 – Descrição formal do caso de uso Compositor de Frases.....</i>	<i>143</i>
<i>Anexo 7 – Descrição dos PageMethods utilizados na aplicação front-end.....</i>	<i>144</i>
<i>Anexo 8 – Resultados da aplicação (algumas screenshots)</i>	<i>146</i>
<i>Anexo 9 – Guião da avaliação do Dicionário.....</i>	<i>149</i>
<i>Anexo 10 – Questionário da avaliação do Dicionário</i>	<i>150</i>
<i>Anexo 11 – Guião da avaliação do modo Adivinhar.....</i>	<i>152</i>
<i>Anexo 12 – Questionário da avaliação do modo Adivinhar</i>	<i>153</i>
<i>Anexo 13 – Questionário Genérico</i>	<i>155</i>

Lista de Figuras

<i>Figura 1 - Modelo com texturas</i>	<i>34</i>
<i>Figura 2- Malha poligonal do avatar.....</i>	<i>34</i>
<i>Figura 3 – Esqueleto do avatar.....</i>	<i>35</i>
<i>Figura 4 – Ator com fato de velcro elástico e marcadores (sistema ótico) (Dinis 2012).....</i>	<i>37</i>
<i>Figura 5 - Microsoft Kinect ®</i>	<i>38</i>
<i>Figura 6 – As 20 articulações detetadas pelo Kinect</i>	<i>39</i>
<i>Figura 7 – 3Gear, teste simples ao finger tracking</i>	<i>39</i>
<i>Figura 8 – FORTH Kinect 3D Hand Tracking, exemplo de tracking</i>	<i>40</i>
<i>Figura 9 - Interface de jogo CopyCat (Lee, Henderson e Brashear 2005)</i>	<i>41</i>
<i>Figura 10 - Interface do Construtor 3D (Deusdado 2002)</i>	<i>42</i>
<i>Figura 11 -- Abecedário 3D exibindo letra “F” (Deusdado 2002).....</i>	<i>43</i>
<i>Figura 12 - Gesto do número 0 no Gestuário 2D (Deusdado 2002)</i>	<i>43</i>
<i>Figura 13- Animações Vídeo exibindo verbo “Amar” (Deusdado 2002)</i>	<i>43</i>
<i>Figura 14 – Equipamento para a captura de movimentos a partir de um intérprete profissional de Língua Gestual. Utilizado nos trabalhos Simon-the-Signer, secção 2.2.4, e 2.2.5, da esq. para a direita (Farzad, et al. 1999, Stephen Cox, et al. 2002, Verlinden, Tijsseling e Frowein 2002).....</i>	<i>44</i>
<i>Figura 15 - Intérprete virtual Simon (Farzad, et al. 1999)</i>	<i>44</i>
<i>Figura 16- Avatar Visia e interface da aplicação (Verlinden, Tijsseling e Frowein 2002)</i>	<i>46</i>
<i>Figura 17 – eSign (Zwitserslood, et al. 2005)</i>	<i>47</i>
<i>Figura 18 - Curso de Língua Gestual: conteúdos do primeiro ao terceiro nível (esq. para a direita).....</i>	<i>48</i>
<i>Figura 19 - Curso em Língua Gestual: exercícios (esq.), tabela de classificação (direita).....</i>	<i>48</i>
<i>Figura 20 – Interfaces Web (esq.) e de Android (direita) do Spread the Sign.....</i>	<i>49</i>
<i>Figura 21 – Signing Savvy: construtor de frases</i>	<i>49</i>
<i>Figura 22 – Signing Savvy: consultar gesto</i>	<i>49</i>
<i>Figura 23 – Signing Science Pictionary</i>	<i>50</i>
<i>Figura 24 – Gestos estáticos para as letras “A” (esq.) e “O” (direita) em LGP no MotionBuilder</i>	<i>53</i>
<i>Figura 25 – Plugin Soletrar palavras em LGP, situado na margem inferior do MotionBuilder.....</i>	<i>54</i>
<i>Figura 26 – Brekel Kinect (captura)</i>	<i>55</i>
<i>Figura 27 – MotionBuilder (captura)</i>	<i>55</i>
<i>Figura 28 - Captura de movimentos integrada num avatar em MotionBuilder</i>	<i>56</i>
<i>Figura 29 – Falha no tracking</i>	<i>57</i>

<i>Figura 30 – Braço trespassa corpo do avatar</i>	<i>57</i>
<i>Figura 31 - Comparação entre animação assistida por Kinect (esq.) e animação manual (dir.)</i>	<i>58</i>
<i>Figura 32 - Poses Animator: interface para carregar no avatar gestos estáticos</i>	<i>59</i>
<i>Figura 33 - Poses Animator: interface para salvar em ficheiro gestos estáticos</i>	<i>59</i>
<i>Figura 34 - Poses Animator: interface para visualizar e editar o conteúdo do ficheiro de gestos</i>	<i>60</i>
<i>Figura 35 - Poses Animator: interface para gerar a animação</i>	<i>60</i>
<i>Figura 36 - Algoritmo de hand tracking por Depth Segmentation</i>	<i>61</i>
<i>Figura 37 - Algoritmo de hand tracking usando as articulações detetadas pelo Kinect.....</i>	<i>62</i>
<i>Figura 38 - Comparação entre os algoritmos de hand tracking desenvolvidos.....</i>	<i>62</i>
<i>Figura 39 – Gestos da letra “A” em LGP com duas perspetivas diferentes</i>	<i>64</i>
<i>Figura 40 - Detecção de contornos para gesto da letra A (esq.) e gesto da letra “F” (direita)</i>	<i>65</i>
<i>Figura 41 - Detecção de contornos para gesto da letra B (esq.) e gesto da letra “D” (direita).....</i>	<i>65</i>
<i>Figura 42- Detecção de contornos para configuração de mão em garra (usada na LGP, lado esq.) e gesto para letra “V” (dir.)</i>	<i>65</i>
<i>Figura 43 - Pose Estimation: ecrã inicial (esq.) e ecrã depois de tirada a foto (direita)</i>	<i>67</i>
<i>Figura 44 - Pose Estimation: interface de marcação de pontos com Zoom.....</i>	<i>67</i>
<i>Figura 45 Pose do gesto mão aberta no MotionBuilder.....</i>	<i>68</i>
<i>Figura 46: Pose</i>	<i>68</i>
<i>Figura 47 - Pose Estimation: marcação de articulações para o gesto relativo à letra “C” em LGP</i>	<i>68</i>
<i>Figura 48 - Pose do gesto para letra “C” em LGP no MotionBuilder</i>	<i>69</i>
<i>Figura 49 - Kinect Skeleton 3D em seated mode com esferas</i>	<i>70</i>
<i>Figura 50 – Kinect Skeleton 3D: full skeleton tracking com cubos.....</i>	<i>70</i>
<i>Figura 51 - Kinect Skeleton 3D: visualização de opções</i>	<i>71</i>
<i>Figura 52 - Kinect Skeleton 3D: esqueleto movido por rotações</i>	<i>72</i>
<i>Figura 53 - Soletrar palavras em LGP com palavras predefinidas (esq.) e Avatar utilizado (dir.)</i>	<i>74</i>
<i>Figura 54 – Menu Story do MotionBuilder (esq.) e avatar utilizado (direita).....</i>	<i>76</i>
<i>Figura 55 – Avatar com problemas na importação.....</i>	<i>80</i>
<i>Figura 56 – Foto dos avatares: (a) Artur - (b) Mimi - (c) Nucha.....</i>	<i>82</i>
<i>Figura 57 – Faceshift: expressões faciais: feliz (esq.), surpreendido (direita), zangado (baixo).....</i>	<i>84</i>
<i>Figura 58 – Faceshift: mapeamento de blendshapes entre avatar e software</i>	<i>84</i>
<i>Figura 59 – Gesto “Olá” em LGP: início do gesto sem expressão facial (esq.) e a meio do gesto vê-se a expressão facial.....</i>	<i>85</i>
<i>Figura 60 – Ver um gesto na interface principal.....</i>	<i>87</i>
<i>Figura 61 - Compositor de frases (em baixo).....</i>	<i>88</i>
<i>Figura 62 - Adivinhar gesto</i>	<i>88</i>
<i>Figura 63 – Gestor de Conteúdos</i>	<i>88</i>
<i>Figura 64 - Caso de uso da Autenticação no sistema</i>	<i>90</i>
<i>Figura 65 - Caso de uso para Traduzir e Adivinhar gestos de palavras</i>	<i>91</i>

Figura 66 - Caso de uso para Compositor de Frases.....	92
Figura 67 - Caso de uso para Backoffice.....	93
Figura 68 – Arquitetura da aplicação	94
Figura 69 – Modelo de Dados do sistema	95
Figura 70 – Diagrama de classes do lado do servidor (front-end).....	96
Figura 71 – Ficheiros javascript usados no lado do cliente (browser)	97
Figura 72 – Protótipo de BF: interface inicial à esq. e interface principal à direita	100
Figura 73 – Protótipo de BF: interface principal iterado à esq. e interface vídeo à direita	100
Figura 74 – Protótipo de BF (iteração sobre o interface compositor de frases)	101
Figura 75 – Interface principal (protótipo de BF à esq. vs. AF à direita).....	101
Figura 76 – Interface para visualizar vídeo (baixa à esq. vs. alta fidelidade à direita).....	101
Figura 77 – Evolução do protótipo de AF para a Lista de Palavras (evolução em cima, versão final em baixo) .	103
Figura 78 – Evolução do protótipo de AF para Botões de Opções (evolução em cima, versão final em baixo) ..	104
Figura 79 – Compositor de Frases (protótipo de AF)	104
Figura 80 – Autenticação (front-end à esq. e backoffice à direita)	106
Figura 81 – divs do layout: logo, contentBar e leftBar	108
Figura 82 – Botões de opções: voltar ao interface inicial, ver definições, e fazer login/logout.....	108
Figura 83 - Lista de palavras.....	109
Figura 84 – Ver Vídeo	110
Figura 85 – Compositor de Frases: arrastando uma palavra	113
Figura 86 – Gráfico da média de respostas ao teste do Dicionário	120
Figura 87 – Gráfico em círculo de respostas do primeiro questionário	121
Figura 88 – Gráfico da média das respostas ao questionário do modo Adivinhar	121
Figura 89 – Gráficos em círculo de respostas do segundo questionário.....	122
Figura 90 – Gráfico da média dos resultados para o Questionário Geral	122
Figura 91 – Interface Login (front-end) Figura 92 – Interface Home (front-end).....	146
Figura 93 - Interface Dicionário, ver gesto (front-end).....	146
Figura 94 – Interface Dicionário: compositor de frases (front-end)	147
Figura 95 - Interface Adivinhar Gesto: início (front-end).....	147
Figura 96 – Interface Adivinhar Gesto, utilizador acertou (front-end)	147
Figura 97 – Interface Login (backoffice)	148
Figura 98 – Interface Principal (backoffice).....	148
Figura 99 – Interface Editar Utilizadores (backoffice)	148
Figura 100 – Interface Ver Utilizadores (backoffice)	148

Lista de Tabelas

<i>Tabela 1 - Comparação entre sensor Microsoft Kinect® e Leap Motion®</i>	<i>40</i>
<i>Tabela 2 – Propriedades principais nos Dicionários Gestuais descritos</i>	<i>51</i>
<i>Tabela 3 – Frases exemplo em LP e respetiva glosa</i>	<i>75</i>
<i>Tabela 4 – Fontes encontradas para definição do interface</i>	<i>102</i>
<i>Tabela 5 – Questionário do teste ao Dicionário</i>	<i>118</i>
<i>Tabela 6 – Questionário do teste ao modo Adivinhar</i>	<i>118</i>
<i>Tabela 7 – Questionário Geral.....</i>	<i>119</i>
<i>Tabela 8 – Descrição formal do caso de uso Traduzir palavram em LG</i>	<i>141</i>
<i>Tabela 9 – Descrição formal do caso de uso Adivinhar gesto para uma determinada palavra</i>	<i>142</i>
<i>Tabela 10 – Descrição formal do caso de uso Traduzir frase em LG</i>	<i>143</i>

Listagens

<i>Listagem 1 - Criação de sessão através de cookie explícito (front-end)</i>	<i>106</i>
<i>Listagem 2 – Criação de sessão e redirecionamento por FormsAuthentication.....</i>	<i>106</i>
<i>Listagem 3 – Lista de PageMethods ou métodos AJAX utilizados no front-end</i>	<i>107</i>
<i>Listagem 4 – Código para fazer uma pesquisa (front-end)</i>	<i>109</i>
<i>Listagem 5 - Código para alternar entre vídeo com loop activou ou inativo.....</i>	<i>110</i>
<i>Listagem 6 - Código de definição de DragSource's e DropTarget's</i>	<i>111</i>
<i>Listagem 7 – Código para processar a ação do utilizador quando tenta arrastar uma palavra para o espaço respetivo do Compositor de Frases.....</i>	<i>112</i>
<i>Listagem 8 – Código para fazer o menino assistente “falar” (em Adivinhar gesto)</i>	<i>113</i>
<i>Listagem 9 – Algoritmo para atribuir o valor do slider à velocidade de vídeo do flowplayer, evitando o comportamento estranho da atribuição direta entre o valor do slider e vídeo.....</i>	<i>115</i>

Capítulo 1

Introdução

Num contexto digital os avatares personificam um utilizador ou agente. São personagens virtuais que representam graficamente entidades que têm usualmente, mas não exclusivamente, aparência humana. Os avatares ou personagens computadorizadas têm vindo a ser aplicados numa panóplia de áreas distintas, nomeadamente, na psicoterapia, indústria cinematográfica e de videojogos, redes sociais, publicidade, entre outras.

Por seu turno, a tecnologia de captura de movimentos (*motion capture*) tem permitido animar estes personagens virtuais de forma mais natural dotando-os de movimentos credíveis. Esta tecnologia utiliza câmeras especiais e cenários próprios para capturar o movimento humano. Este trabalho focou-se numa área particular de aplicação dos avatares: um intérprete virtual para expressar Língua Gestual.

Este projeto foi realizado no âmbito da cadeira de Projeto de Engenharia Informática no DI-LabMAG (Departamento de Informática- Laboratório de Modelação de Agentes).

1.1 Motivação

“Os media eletrónicos, tais como a televisão, filmes e Internet são fontes de valor incalculável de informação e entretenimento, mas para um número significativo de pessoas, as deficiências auditivas tornam difícil o desfrutar do mundo digital que a maioria toma como garantido”.¹

Existem alguns métodos para tornar acessível a pessoas surdas, a informação divulgada no mundo eletrónico. De todos os métodos, o mais utilizado é a legendagem². A leitura de legendas é muito útil para pessoas com dificuldades auditivas ou que ficaram surdas ao longo da vida. No entanto, as pessoas surdas congénitas (à nascença) têm dificuldades em compreender a língua escrita, e acham muito difícil interpretar as legendas. Esta situação deve-se ao facto de o surdo

¹ Transcrição livre de (Power e Power s.d.)

² A **legendagem** consiste na transcrição de discurso sonoro em texto.

congénito ter mais dificuldade do que uma pessoa ouvinte em estabelecer a relação entre as palavras escritas e o sentido de abstração das mesmas (Power e Power s.d.). Para a maioria dos surdos, a língua gestual é a sua língua materna, ou seja, uma vez que é a sua primeira língua, é a língua que usam naturalmente. Esta língua é bastante diferente de uma língua oral, tendo um léxico e uma gramática próprios. Os intérpretes qualificados de Língua Gestual escasseiam pelo que faz sentido recorrer a personagens virtuais ou avatares.

A computação gráfica tem vindo a evoluir ao longo dos anos. A qualidade gráfica dos avatares está continuamente a aproximar-se de uma representação cada mais vez realista de humanos verdadeiros. Este cenário cria a oportunidade para aplicações com personagens mais expressivas, mais apelativas, qualidades determinantes para a usabilidade e aceitação da aplicação por parte do público-alvo. Uma das vantagens do uso de avatares em relação ao vídeo é a fácil manipulação do avatar e do ambiente 3D onde está inserido.

Tradicionalmente, a expressão da Língua Gestual em formato eletrónico faz-se recorrendo ao vídeo. Existem vantagens e desvantagens no uso de avatares em detrimento do vídeo. A desvantagem é o tempo de produção da animação do avatar em comparação com a rapidez de gravação de um gesto em vídeo. Todavia, a principal vantagem é que a animação de um gesto pode ser manipulada e reutilizada muito mais facilmente do que um vídeo. Estas características permitem formar frases através da combinação de vários gestos atómicos (i.e. gestos de palavras), permite também a troca de intérprete virtual por outro distinto, ou até mesmo à alteração das suas roupas e adereços, enquanto para o vídeo isto não é possível sem ser refeito ou filmado de novo. Uma vez que os avatares residem num ambiente virtual 3D, vão de encontro à natureza tridimensional da própria Língua Gestual, logo, este ambiente torna possível a visualização e compreensão do gesto em ângulos diferentes. Assim abrem-se portas a aplicações mais interativas e a novos cenários de aplicações em relação ao uso do vídeo.

Até há bem pouco tempo, a captura de movimentos era suportada exclusivamente por uma tecnologia bastante cara. Sendo necessário equipamento dispendioso, espaço próprio, recursos humanos especializados e caros para operar uma sessão de captura. O aparecimento recente de novas tecnologias, com especial destaque para o Microsoft Kinect[®], muito menos dispendiosas, veio tornar acessível a captura de movimentos num ambiente vulgar e produzir animações para um avatar com movimentos mais realistas do que a tradicional animação 3D.

1.2 Objetivos

Este projeto tem como objetivo fundamental conceber um avatar animado que personifique um intérprete virtual da Língua Gestual Portuguesa (LGP). As etapas necessárias são a geração do modelo do avatar, passando pela animação do próprio com gestos de Língua Gestual Portuguesa, e desenvolver uma aplicação que suporte a interação com este personagem virtual.

O conjunto de gestos produzido neste trabalho será representativo de uma prova de conceito, e obtido por um processo de animação assistido por uma tecnologia de captura de movimentos de baixo-custo. Pretende-se também chegar à combinação de gestos atômicos de modo a formar frases simples em LGP.

1.3 Contribuições / Trabalho realizado

Neste trabalho foram usados vários avatares, um dos quais considerado adequado por técnicas da Associação Portuguesa de Surdos para executar gestos de LGP no âmbito da interação com crianças.

Foi realizada animação de cerca de 35 gestos que correspondem a palavras de LGP usando um método híbrido, por um lado a captura do movimento do corpo com o sensor de profundidade Microsoft Kinect[®], e por outro, recorrendo a animação manual para descrever a configuração da mão do gesto. Foram produzidos gestos para representar pronomes pessoais, pronomes interrogativos, verbos, substantivos, entre outros. A partir deste conjunto foram feitas algumas combinações de gestos atômicos de modo a produzir animações de frases em LGP.

De forma a demonstrar as funcionalidades acima conseguidas, foi criada uma aplicação *Web* denominada *Os meus primeiros gestos*, que permite consultar, procurar palavras, ver a tradução do seu significado em LGP, compor frases selecionando palavras ou gestos atômicos, e testar os conhecimentos do utilizador através de um modo para adivinhar palavras. De modo a estender as funcionalidades da aplicação anterior, foi desenvolvida outra aplicação *Web* do tipo *Backoffice* ou gestor de conteúdos, destinada a superutilizadores para gerir os conteúdos da aplicação principal.

No desenvolvimento desta tese também resultaram como investigação e trabalho preliminar, as seguintes ferramentas de *software*:

- Dois *plugins* para a aplicação *MotionBuilder* (Autodesk 2011): o primeiro denomina-se *Soletrar Palavras em LGP*, converte qualquer letra introduzida por um utilizador para um gesto de LGP; o segundo denomina-se *Poses Animator* e serve para carregar num avatar gestos estáticos de LGP (i.e. gestos de letras) que tenham sido armazenados previamente num ficheiro no formato próprio da aplicação, além de que consegue produzir uma animação em tempo real de transições de entre vários gestos estáticos armazenados em ficheiro.
- Duas aplicações *Desktop*: a primeira faz o seguimento (*tracking*) automático da pose corporal, juntamente com o *tracking* assistido pelo utilizador da pose 3D dos dedos, e por conseguinte, converte essa informação capturada para pose de um esqueleto em *MotionBuilder*; a segunda permite visualizar um esqueleto a mover-se conforme os movimentos feitos por um utilizador através do sensor Kinect.

- Dois algoritmos de *hand tracking*³ através do sensor Kinect: o primeiro utiliza *Depth Segmentation* para detetar as mãos, e o segundo utiliza a posição das articulações das mãos fornecida diretamente pelo sensor.

1.4 Metodologia de desenvolvimento

A metodologia usada neste projeto foi o processo de desenvolvimento iterativo (DoIT 2005).

As iterações ao longo do projeto foram:

1. Desenvolvimento de um protótipo composto por um *plugin* para Soletrar Palavras em LGP (3.3) e por um subconjunto de frases em LGP feitas a partir da combinação de gestos atômicos.
Visita à APS de modo a avaliar o protótipo desenvolvido (4.1.2).
2. Criação de um novo avatar e a sua reavaliação por surdos, com o intuito de chegar a um personagem virtual mais adequado ao utilizador (4.2).
3. Criação de uma aplicação *Web* que apresente conteúdos em LGP (Capítulo 5).
Teste da aplicação com utilizadores (Capítulo 6).

A principal vantagem desta metodologia é envolver o utilizador no processo de desenvolvimento. Em vez de esperar até que a aplicação se torne num produto final, fase em que cada alteração tem um impacto muito grande, os problemas foram identificados e resolvidos nas fases intermédias. Problemas como rigor insuficiente na animação, escolha de avatar inadequado para o objetivo, entre outros. Outra vantagem deste método foi que em cada iteração as metas do projeto foram reavaliadas, conforme o *insight* obtido em cada iteração.

1.5 Estrutura do documento

Este documento é composto por 7 capítulos, estruturados da forma que a seguir se apresenta.

O **capítulo 1** apresenta uma vista geral do trabalho aqui descrito, bem como os objetivos principais e as razões que motivaram este projeto.

O **capítulo 2** começa por apresentar um conjunto de conceitos relevantes ao entendimento dos temas abordados na tese. Em seguida, apresenta uma perspetiva geral do estado da arte, relatando os projetos relacionados com as áreas de Língua Gestual e Avatares.

³ *Hand tracking* consiste na determinação da posição da mão em cada instante de tempo.

No **capítulo 3** apresenta-se o trabalho preliminar realizado para dominar a tecnologia utilizada. Neste capítulo apresenta-se toda a investigação e testes efetuados nas áreas de animação, captura de movimentos, *hand tracking* e *finger tracking*.

O **capítulo 4** descreve o processo de geração e animação do avatar, na componente corporal e facial. Também se apresentam os resultados dos primeiros testes com utilizadores.

No **capítulo 5** apresenta-se a aplicação desenvolvida onde se descreve detalhadamente o processo realizado na sua construção.

No **capítulo 6** descrevem-se os testes com utilizadores realizados à aplicação desenvolvida.

Por último, no **capítulo 7** faz-se a conclusão do trabalho, resumindo os principais temas abordados na tese e descrevendo os caminhos que o projeto pode tomar no futuro.

Capítulo 2

Conceitos e Trabalho Relacionado

Este capítulo apresenta os conceitos fundamentais do contexto desta tese, bem como alguns trabalhos relacionados. Os primeiros trabalhos apresentados consistem em projetos de referência que cruzam as áreas de Avatares e Língua Gestual. Em seguida destaca-se um tipo particular de projetos, os dicionários de gestos, pois estes representam o grupo de aplicações que mais se aproxima da aplicação final desta tese. O capítulo é concluído com uma comparação entre os trabalhos deste grupo, confrontando-se algumas das características mais importantes de cada um.

2.1 Conceitos

2.1.1 Língua Gestual Portuguesa

A maioria dos países com um número significativo de surdos congénitos tem uma língua gestual própria. Em Portugal⁴ existe a Língua Gestual Portuguesa (LGP), no Brasil existe a Língua Brasileira de Sinais (LIBRAS) e nos Estados Unidos existe a *American Sign Language* (ASL), referindo apenas alguns exemplos.

No contexto da língua gestual, define-se por gesto, um movimento ou sequência de movimentos necessários à representação de uma ideia. A execução de um gesto pode ser decomposta em três componentes: movimento dos membros superiores, configuração e orientação das mãos, e expressão facial. O indivíduo que **gestua**⁵, isto é, que executa gestos, terá de ter sempre uma mão dominante (usualmente a direita), a sua mão principal e mais natural à realização a língua gestual.

As línguas gestuais não são reproduções da língua escrita e falada. Cada língua gestual utiliza expressões faciais e movimento corporal para criar uma linguagem própria: com características sintáticas, gramaticais e semânticas, idênticas às das línguas faladas. A língua gestual impõe

⁴ Para mais informações sobre o contexto socio-cultural dos surdos portuguesa consultar o *Anexo I*.

⁵ A palavra *gestua* (verbo *gestuar*) é um termo específico que significa transformar em gesto.

também que a representação contextual dos sentimentos, pensamentos e do raciocínio seja transmitida, e não só as frases e palavras.

Em LGP, uma das formas de expressar os sentimentos é pela adição dos gestos “*triste*”, “*contente*” ou “*aborrecido*” à frase, para nomear alguns. No entanto, a expressão dos sentimentos é representada principalmente pelo uso e reforço do gesto com expressões faciais. Por exemplo, a expressão facial do sorriso associado à frase em Língua Portuguesa (LP) “*Eu passei no exame de código*” transmite uma ideia positiva e de felicidade. A intensidade da expressão facial também define o nível de entoação que se quer dar ao sentimento. A expressão facial define o sentido emocional da mensagem, assim como também, serve para distinguir gestos em que a configuração manual do gesto (i.e. movimento dos braços e configuração da mão) seja igual, mas que tenham significados diferentes. Vejamos a frase em LP, “*O meu irmão chama-se João*”, em LGP o gesto para “*irmão*” se for acompanhado de expressão da boca “*som papa*”, representa a frase referida. Caso o mesmo gesto seja acompanhado com a expressão “*bochecha direita cheia*”, então corresponde à palavra “*igual*”.

Muitas vezes uma palavra tem uma correspondência direta com um gesto, porém, isso nem sempre acontece. A palavra “*médico*” surge da combinação do gesto “*doente*” seguido do gesto “*auscultar*”. Nesse caso, considera-se “*médico*” um gesto não atômico, composto pelos gestos atômicos “*doente*” e “*auscultar*”.

A glosa passa para um registo escrito a forma como a estrutura da LGP se organiza. Os exemplos em seguida que traduzem textualmente os gestos de LGP usam este método. A LP tem como estrutura frásica sujeito-verbo-objeto (SVO), por exemplo, consideremos a frase: “*Eu vou à escola*”, onde “EU” é o sujeito, “VOU” o verbo, e “ESCOLA” o objeto. A LGP tem dois tipos de estrutura frásica possíveis: sujeito-objeto-verbo (SOV) e objeto-sujeito-verbo (OSV). A frase anterior ficaria “*ESCOLA EU IR*” (glosa), usando sintaxe OSV. O verbo aparece no infinitivo e no fim da frase. As preposições, os artigos definidos e indefinidos são omitidos na língua gestual portuguesa.

Quem se expressa em língua gestual, especialmente um surdo congénito, pensa “visualmente” e gestua diretamente da forma como pensa. Na prática, a ordem das palavras reflete a importância que se quer dar a cada uma. Consideremos um exemplo, no qual um sujeito quer dizer a alguém que vai ao cinema e está muito entusiasmado. A primeira coisa que o sujeito pensa é no cinema e depois em ir para lá. Então o sujeito gestuará: “*CINEMA EU IR*” (OSV). Por outro lado, se o sujeito estiver sereno em relação a um passeio, e ocorrer-lhe a ideia de ir ao cinema, geralmente gestuaria: “*EU CINEMA IR*” (SOV).

Conceitos gramaticais

A marcação do género é realizada somente nos seres animados. É utilizado o prefixo “*mulher*” para marcar o género feminino, enquanto o masculino geralmente é caracterizado pela

ausência da marca “*homem*”. Para exemplificar o feminino, considere-se a palavra “*rainha*”, esta é traduzida pelo gesto “*mulher*” seguido do gesto “*rei*”. Todavia existem exceções, por exemplo, palavras como “*pai*” / “*mãe*”, ou “*carneiro*” / “*ovelha*”, têm um gesto distinto para cada gênero.

A concordância de número pode ser realizada por várias maneiras: repetição do gesto; por redobro (i.e. realização do gesto por ambas as mãos); incorporação de um número (por ex.: “LIVRO CINCO”); ou por incorporação do sufixo “*muitos*”. Normalmente, nestas situações os gestos são acompanhados de reforço da expressão facial.

Para a marcação do tempo verbal é utilizado por sufixação o gesto de “*passado*” ou “*futuro*” à forma infinitiva do verbo, marcando o respetivo tempo verbal.

Os tipos de frase interrogativa e exclamativa requerem o uso da expressão facial durante toda a frase. No caso da interrogativa, esta pode ser combinada com pronomes interrogativos que surgem no final da frase. Como exemplo, a frase em LP “*Onde Vais?*”, ficaria em LGP “TU IR ONDE?”. Quanto à marcação da forma negativa, pode-se acrescentar o gesto “*não*” no fim da frase, ou realizar-se simultaneamente a frase e o movimento natural da cabeça referente a “*não*”.

Português Gestual

O Português Gestual (PG) utiliza o vocabulário da LGP numa estrutura frásica igual à da LP. Serve unicamente para mostrar a ordem dos gestos com a sintaxe do Português, e não transmite de maneira válida o pensamento de um surdo congénito. O PG é considerado um sistema artificial, destinada a ouvintes, ao contrário da LGP que é uma língua natural. Em cada língua gestual existe este sistema equivalente, por exemplo, no Reino Unido, existe o *Sign Supported English* (SSE), que utiliza o vocabulário da língua gestual britânica com a estrutura frásica da língua inglesa.

2.1.2 Avatar

A palavra “avatar” tem origem no Hinduísmo e serve para designar a representação terrestre de uma divindade. Em computação, avatar consiste na representação gráfica do utilizador ou alter-ego deste. Certos autores (Damer, Judson e Dove 1999) têm uma definição mais ampla de avatar e definem-nos como “personagens, jogadores, atores virtuais, ícones ou seres humanos virtuais”. Um avatar pode ter ou não aparência humana, e ter uma forma tridimensional como nos jogos de computador, ou uma forma bidimensional comum nos fóruns da Internet. Nos mundos virtuais, nomeadamente nos jogos, o avatar é um personagem interativo. Em muitos casos, todos os aspetos do personagem podem ser parametrizados como o tipo de musculatura, cabelo, pele ou roupa. Um exemplo de um mundo virtual repleto de avatares é o jogo *Second Life* (da Silva 2010).

Para percebermos como se anima um avatar inserido num mundo virtual, é preciso primeiro compreender a estrutura interna de um personagem virtual. Os parágrafos seguintes apresentam o conceito de avatar no contexto da animação.

Um avatar é um modelo gráfico, cuja representação gráfica assenta numa malha poligonal. Esta malha, também conhecida como *mesh*, representa uma estrutura modelada a partir de formas elementares geométricas. No final da modelação da *mesh* obtém-se a superfície que dá forma à personagem (*Figura 1*). Em seguida é necessário representar o seu aspeto exterior, aplicando materiais.

Os materiais são elementos que representam as propriedades de cor, intensidade de reflexo, sombra, transparência, etc. Um exemplo de material é o vidro de um copo. Este material é definido pelo conjunto de todas as propriedades óticas que fazem o vidro ter um aspeto transparente e reflexivo. Na *Figura 1*, vemos o resultado da aplicação dos materiais à malha ou superfície do personagem. Uma vez que a natureza de um material é plana, é necessário um mapa de texturas que molde o material no plano 2D para a fisionomia tridimensional do objeto alvo. Este mapa de texturas consiste num mapa de coordenadas que a partir de uma imagem 2D “envolve” o objeto 3D.

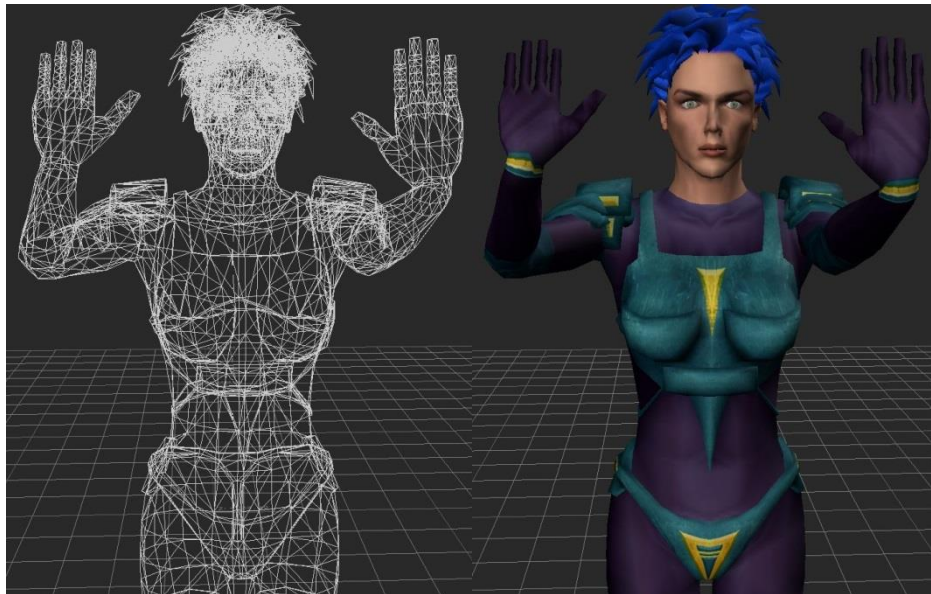


Figura 2- Malha poligonal
do avatar

Figura 1 - Modelo com
texturas

De modo a que um avatar se consiga mover, é necessário criar o seu esqueleto. O esqueleto é composto por um conjunto de ossos e articulações. As articulações são os vértices ou pontos suscetíveis de rotações e translações nos vários eixos. À ligação dos vértices entre si, chamam-se segmentos rígidos ou ossos. O conjunto de articulações forma uma estrutura hierarquizada (*Figura 3*), onde cada articulação conhece, caso se aplique, a sua articulação pai e filho. Por exemplo, tanto para o lado esquerdo como direito, a articulação do cotovelo tem como pai a articulação do ombro, e o pulso como articulação filho.

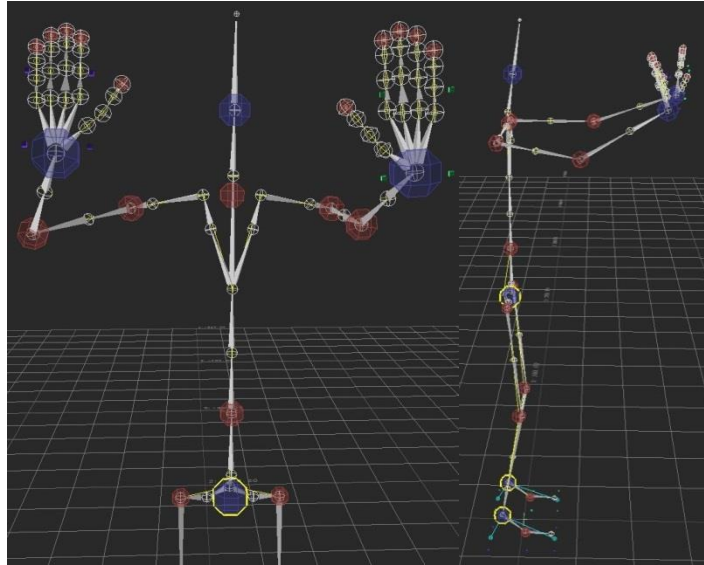


Figura 3 – Esqueleto do avatar

O processo seguinte à criação de um esqueleto chama-se *skinning*. Esta operação tem como objetivo associar os ossos à geometria da *mesh*, isto faz com que à medida que o esqueleto se mova, a pele do avatar acompanhe o movimento.

Animação e síntese de gestos

A animação de um avatar consiste na definição dos valores de rotação e translação (nos vários eixos), para cada articulação em cada instante de tempo, dando a perceção de movimento. Existem pelo menos dois modos para animar um avatar.

O primeiro modo consiste na forma tradicional de animação manual por *keyframe*. O movimento é produzido com base na criação de várias posturas ou poses distintas, distribuídas ao longo de instantes temporais. O *software* utilizado para gerar o movimento, automaticamente produz a animação através de uma função de interpolação. Esta, é responsável por preencher as posições de cada articulação para todas as *frames* entre uma dada pose inicial e final.

O segundo método consiste na utilização da técnica de captura de movimentos descrito mais adiante, em 2.1.3 .

A síntese de gestos é o processo de produzir uma animação que contempla todas as componentes de movimento que caracterizam um gesto, designadamente, o movimento dos membros superiores, das mãos, dedos, e a expressão facial. Tecnicamente existem duas formas de sintetizar um gesto.

A primeira é baseada na correspondência direta entre os significados de palavras existentes numa língua oral e os gestos de uma língua gestual. Neste caso, é armazenado o significado da palavra de uma dada língua oral, fazendo-lhe equivaler um identificador de gesto. Esse identificador corresponde a uma animação. Dessa forma, quando escrevemos uma palavra na

língua oral que esteja armazenada, o avatar gestua o equivalente na língua gestual. Este método é utilizado nos projetos Simon-the-Signer, TESSA e Visia apresentados no trabalho relacionado.

A segunda baseia-se numa análise linguística do gesto. Nestes casos, é realizada uma análise paramétrica dos elementos constituintes de cada gesto. Esses elementos são depois armazenados, numa entrada correspondente ao gesto, o que pode incluir o equivalente numa língua oral, designadamente uma palavra. No geral, os parâmetros analisados são a configuração, local da articulação e da orientação das mãos, movimento corporal e expressão facial. Deste modo, pode-se gerar os movimentos do avatar diretamente a partir da descrição linguística, ou seja, o sistema compõe o gesto a partir da soma das instruções do valor de cada parâmetro (Martins, Ferreira e Mineiro 2012). Este foi o método utilizado pelo projeto eSign, descrito mais à frente no estado da arte.

Animação facial

O processo de animação facial é usualmente realizado por *Blendshapes* ou *Facial Bone Rig*. O último recorre à criação dos ossos da face que mais tarde são animados através da definição dos valores de translação e rotação em cada osso. A grande vantagem deste método é que a mesma animação pode ser utilizada para personagens distintas, desde que a fisiologia facial seja similar.

A técnica de animação por *blendshapes* utiliza uma série de *morph targets* pré-estabelecidos. Cada *morph target* armazena uma versão deformada da *mesh* com a expressão facial definida. Durante a animação, a pose da cara na posição neutra é interpolada com a posição dos vértices definidos por cada *morph target*. Geralmente, os valores de cada *morph target* vão de 0 a 1. Este método apresenta um nível mais elevado de fidelidade nas expressões faciais do que o método anterior. No entanto, envolve muito mais trabalho manual na criação dos *morph targets*, e que por sua vez, são específicos a cada personagem.

2.1.3 Captura de movimentos

A captura de movimento ou *motion capture*, consiste no processo de aquisição do movimento e tradução do mesmo para um modelo digital bi ou tridimensional (Dinis 2012). O objeto de captura pode ser direcionado a humanos, animais ou qualquer objeto a que se possa adicionar sensores. Os dados adquiridos podem ser focados para o estudo e análise do movimento, bem como para a criação de animações digitais.

Esta técnica é aplicada a uma grande diversidade de áreas, nomeadamente, a aplicações médicas, militares e de entretenimento. No desporto, o uso de *motion capture* tem interesse para analisar o movimento de atletas com o objetivo de aumentar o seu rendimento desportivo e prevenir eventuais lesões. Na ergonomia, a tecnologia é utilizada com vista ao desenvolvimento de produtos mais adequados e confortáveis ao público-alvo. É também aplicado ao entretenimento como por exemplo: no cinema, no filme “*Avatar*” de James Cameron, onde se investigou um novo

método mais realista para capturar as expressões faciais; em jogos de computador como “*Grand Theft Auto IV*”, “*Guitar Hero*”; nomeando apenas alguns exemplos.

A tecnologia (*Figura 4*) permite capturar a posição dos membros do corpo do indivíduo, até informações mais complexas como movimento das mãos, dedos ou movimentos faciais. No geral, capturam-se separadamente os movimentos corporais dos movimentos faciais, devido às circunstâncias e subtilezas dos últimos (Dinis 2012). O termo *hand-over* refere-se ao processo de adição da animação de dedos e mãos, a uma captura já realizada para o corpo através de, por exemplo, uma luva de captura de movimentos (Measurand Inc. 2011).

O *motion capture* é um processo complexo e composto por vários passos sequenciais. Os seguintes passos são usados numa captura com um sistema ótico (Nagle 1999):

1. Preparação do estúdio;
2. Calibração dos sensores e volume da captura;
3. Captura ou gravação do movimento;
4. Tratamento de dados;
5. Pós-produção dos dados, ou seja, aplicação dos dados a objetos bi ou tridimensionais.

As vantagens principais desta técnica são a rapidez no processo de animação e o resultado realista que traz às animações em detrimento da animação manual por *keyframe*. A sua maior desvantagem é ser dispendioso em termos de equipamento, *software* e pessoal especializado.

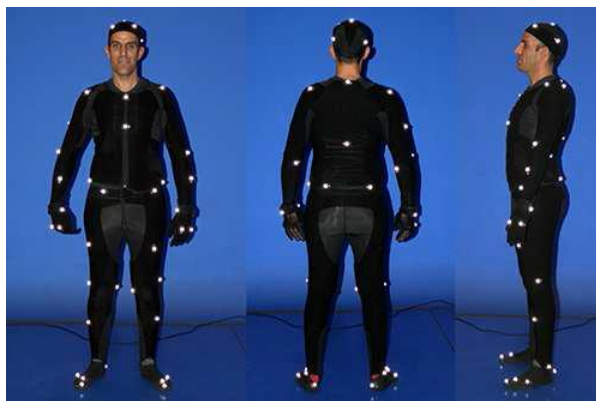


Figura 4 – Ator com fato de velcro elástico e marcadores (sistema ótico) (Dinis 2012)

2.1.4 Sensores de profundidade

Os sensores de profundidade estabelecem a relação entre um ponto da imagem, e a distância a que este se situa do sensor. Por outras palavras, o sensor de profundidade dá-nos a coordenada Z do espaço tridimensional, em complemento, às coordenadas X e Y já obtidas pelas câmeras tradicionais de cor *RGB*. Esta tecnologia apresentou o processamento de dados de profundidade de uma forma muito inovadora do qual drasticamente cortou o preço de sistemas que até à altura

faziam o reconhecimento de profundidade, nomeadamente, por câmeras *Time of Flight* ou câmeras estereoscópicas. Com isto, estas câmeras vieram abrir portas à investigação de algoritmos modernos para realizar a captura de movimentos, tanto a nível corporal, como das mãos (*hand tracking*), dos dedos (*finger tracking*), assim como também a nível facial.

Microsoft Kinect[®]

Microsoft Kinect[®] é uma câmara lançada em 2010 originalmente apenas como acessório da consola de jogos Microsoft Xbox[®]. A câmara (*Figura 5*) é capaz de fazer o reconhecimento: das articulações do utilizador, reconhecimento facial; e reconhecimento de voz. Mais tarde em 2012 foi lançado uma versão específica para PC. O Kinect está equipado com uma câmara de cor *RGB*, um sensor de infravermelhos, um emissor de luz infravermelha, e quatro microfones.



Figura 5 - Microsoft Kinect[®]

O sistema é capaz de reconhecer 20 articulações (*Figura 6*) do corpo humano e além disso, tem suporte para a captura das expressões faciais. Todavia, não suporta a deteção direta da posição dos dedos, essencial à definição da configuração da mão num gesto. Este problema tem de ser abordado a nível de *software*, através de um algoritmo que faça *finger tracking*. Para consultar as especificações técnicas e a descrição do algoritmo inovador usado por este sistema consultar o *Anexo 2*.

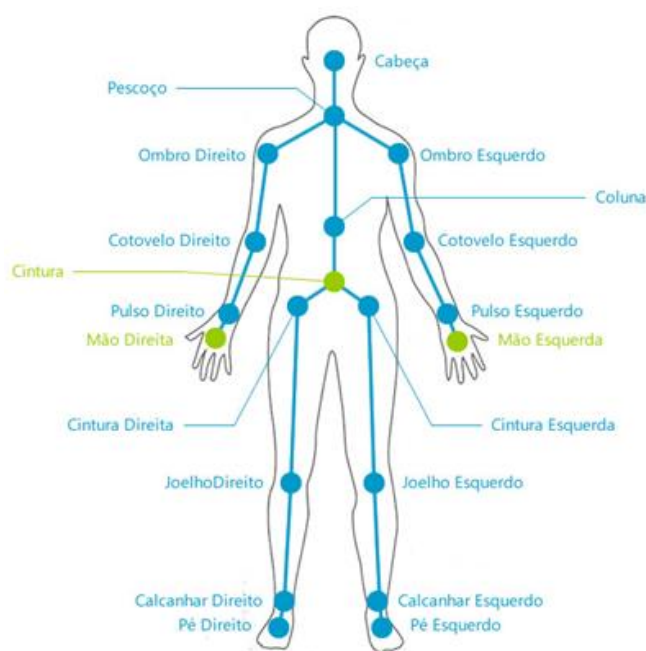


Figura 6 – As 20 articulações detetadas pelo Kinect

Aplicações testadas

3Gear: A aplicação 3Gear (Wang, Twigg e Kin 2012) é um *SDK (System Development Kit)* que suporta a implementação de algoritmos de *hand* e *finger tracking*. O *software* foi desenhado para fazer o *tracking* tendo uma vista superior das mãos do utilizador, estando direcionado para a secretária deste e não para a sua frente. Uma vez que os gestos em Língua Gestual são feitos de frente para o recetor, este algoritmo não se adapta aos gestos de LGP. A outra limitação é que este *software* pretende detetar apenas a ponta dos dedos, excluindo o resto das suas articulações necessárias à definição do gesto. Pela avaliação desta aplicação, conclui-se que o sistema não era suficiente para suportar o *tracking* dos dedos inerentes aos movimentos gestuais. Na *Figura 7*, do lado esq. está representado o modelo virtual da mão, e no lado direito, estão pintas com cores distintas cada mão detetada pelo algoritmo.

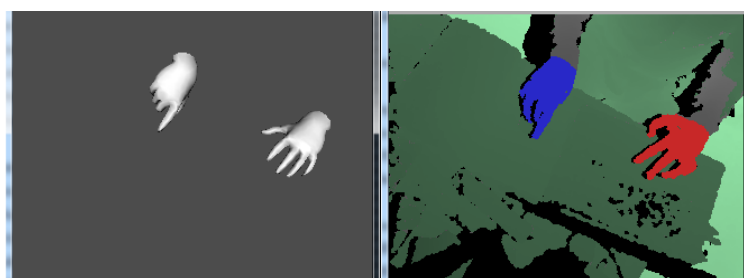


Figura 7 – 3Gear, teste simples ao *finger tracking*

FORTH: O sistema seguinte (Oikonomidis, Kyriazis e Argyros 2011) é uma API que permite detetar 26 níveis de liberdade na articulação da mão. O sistema requer uma placa gráfica *NVIDIA* para funcionar, e apesar dos requisitos da máquina onde se testou o sistema, serem superiores aos

requisitos mínimos apontados, não foi possível testar a aplicação dada a taxa de imagens por segundo ser inferior a 1. Pelas observações feitas nos vídeos de demonstração, o FORTH faz o *tracking* para a mão e dedos muito bem, sem perder o *tracking*, mesmo com dedos oclusos (Figura 8). No entanto, verificou-se que o utilizador faz sempre gestos muito lentos, sugerindo que o *tracking* se perderia se aumentasse a velocidade do movimento. Dadas as observações e as limitações técnicas encontradas, este sistema também não pode ajudar a desenvolver a solução para uma captura de movimentos ao nível dos dedos

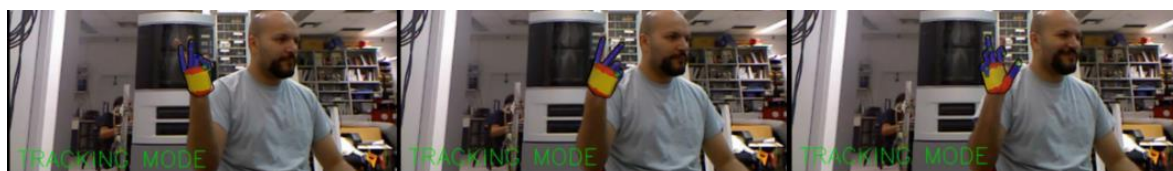


Figura 8 – FORTH Kinect 3D Hand Tracking, exemplo de *tracking*

Leap Motion[®]

O Leap Motion[®] é focado para a captura do movimento das mãos e dedos do utilizador. É uma pequena unidade desenhada para ser posicionada em baixo e à frente do monitor. Tem um alcance de 22 cm². Em seguida, mostra-se na

Tabela 1 uma comparação entre as características deste sensor e do sensor Microsoft Kinect[®].

	Microsoft Kinect [®]	Leap Motion [®]
Vantagens	<ul style="list-style-type: none"> • Detecção do esqueleto e face • Acesso aos dados de profundidade 	<ul style="list-style-type: none"> • Detecção de dedos • Muito pequeno e mais barato
Desvantagens	<ul style="list-style-type: none"> • Precisão baixa nos dados de profundidade ao nível dos dedos 	<ul style="list-style-type: none"> • Alcance curto • Não tem detecção do esqueleto nem face • Sem acesso aos dados de profundidade

Tabela 1 - Comparação entre sensor Microsoft Kinect[®] e Leap Motion[®]

Durante os testes com as aplicações já incluídas no *SDK*, conclui-se que apesar de ter uma boa detecção de dedos não resolve a problemática das oclusões entre eles, perdendo a localização do dedo quando existe uma oclusão ou quando se faz o movimento de adução dos dedos. Torna-se um sistema inviável na sua aplicação à língua gestual pois todos os gestos têm configurações de dedos que provocam a oclusão de outros, ou que fazem o movimento de adução. Outra limitação do sistema, semelhante à encontrada no sistema 3Gear, é que ambos se focam na detecção das pontas dos dedos, e não têm em conta as restantes articulações dos dedos.

Portanto, a minha escolha tecnológica recaiu sobre o sensor Microsoft Kinect[®], porque além de ter *tracking* do esqueleto e suporte ao *tracking* das expressões faciais, permite o acesso aos dados de profundidade, tornando possível desenvolver algoritmos de *tracking* com base nesses dados.

2.2 Trabalhos Relacionados

2.2.1 CopyCat

CopyCat (Lee, Henderson e Brashear 2005) é um jogo interativo em que o utilizador controla uma personagem através da realização de gestos de Língua Gestual Americana (ASL). O jogo (*Figura 9*) é desenhado para crianças surdas onde se pretende que estas melhorem as suas capacidades linguísticas através da prática de ASL. Primeiro, o utilizador assiste a um vídeo de um intérprete humano a executar gestos em ASL, e por sua vez, o utilizador tem de imitar os gestos que visualizou.



Figura 9 - Interface de jogo CopyCat (Lee, Henderson e Brashear 2005)

A interface do jogo é composta por: (A) vídeo tutor; (B) vídeo ao vivo do utilizador; (C) botão para ligar/desligar reconhecimento do gesto; (D) ambiente de jogo; (E) escolha de nível.

O jogo tem duas versões: uma versão inicial que usa uma câmara de vídeo e duas luvas coloridas com acelerómetros (Lee, Henderson e Brashear 2005); e outra versão que usa apenas o sensor Microsoft Kinect (Zafrulla, et al. 2011), lançada seis anos depois. As principais características na utilização do sensor Kinect em detrimento das luvas são: aumento da usabilidade

(i.e. conforto, melhor interação); custo mais acessível; e um decréscimo na precisão do reconhecimento de gestos.

Na versão Kinect do sistema, os elementos que se recolhem para identificar o gesto, consistem nas características da pose corporal e da silhueta da mão. A extração das características da pose corporal, usa as posições tridimensionais das articulações do tronco superior, fornecidas pelo Kinect. Quanto à silhueta da mão, a abordagem utilizada não contempla o *finger tracking*. Ao invés, é usada a posição tridimensional de cada ponto que se encontre na vizinhança da articulação da mão, detetada pelo Kinect.

2.2.2 Ensino da Língua Gestual Assistido por Personagens 3D Virtuais

Este trabalho (Deusdado 2002) tem por objetivo apoiar a comunidade surda com ferramentas tecnológicas com vista à disseminação da LGP. O autor criou uma aplicação gráfica tridimensional *online* em que utiliza um humano virtual, cujo papel é servir de tutor gestual. A aplicação foi desenvolvida usando o *Pulse3D*. Esta tecnologia permite a visualização de conteúdo 3D na *Web* à custa da instalação de um *plugin* para o *browser*, sendo capaz de criar ambientes virtuais interativos a três dimensões.

A aplicação pretende ensinar a LGP de várias formas, e por isso, é composta por várias ferramentas, tendo cada uma um objetivo próprio. As ferramentas criadas foram: *Construtor 3D*; *Abecedário 3D*; *Animações Vídeo*; e *Gestuário 2D*.

O Construtor 3D (*Figura 10*) permite a construção interativa de gestos de LGP. O utilizador visualiza três modelos gráficos: um humano virtual completo, uma mão esquerda e uma mão direita. Cada modelo dispõe de manipuladores tridimensionais para rotações e translações de forma a compor o gesto. Cada gesto criado pode-se associar a uma palavra, letra ou número. Esta ferramenta pretende ajudar na memorização de gestos, implícito no processo de construção virtual, e pretende uniformizar o vocabulário gestual, uma vez que cada gesto construído será eventualmente associado univocamente a uma palavra.

O autor não teve uma grande preocupação com o realismo do modelo virtual, logo, o trabalho focou-se em obter modelos gráficos simples e leves.

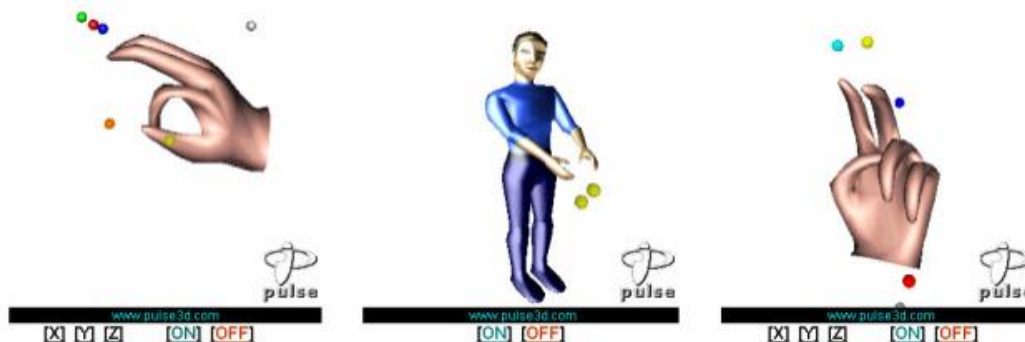


Figura 10 - Interface do Construtor 3D (Deusdado 2002)

O Abecedário 3D (Figura 11) é uma ferramenta que permite a visualização interativa de animações a três dimensões para todas as letras do abecedário. Com esta ferramenta o utilizador observa o modelo do braço e mão de um humano virtual, com a possibilidade de ver a animação de cada letra sobre quatro pontos de vista diferentes: frente, cima, do ponto de vista do humano virtual e uma vista rotativa sobre o braço. Pretende facilitar a memorização de gestos para as letras do abecedário da LGP.

A terceira ferramenta chama-se Animações Vídeo (Figura 13), e tem como foco a visualização de animações 3D dos verbos da LGP. Ao contrário do Abecedário 3D, estas animações não são interativas, de forma que não se pode ver a animação de vários pontos de vista. No entanto, pode-se ver um humano virtual completo em vez de apenas o braço e a mão.

O Gestuário 2D (Figura 13) consiste numa biblioteca de imagens de gestos, nomeadamente dos gestos relativos à numeração. As imagens são acompanhadas de descrições verbais com o maior detalhe possível na informação descritiva do gesto, ou seja, descreve a orientação, localização e os movimentos necessários para definir o gesto.



Figura 11 -- Abecedário 3D exibindo letra “F” (Deusdado 2002)

Figura 13- Animações Vídeo exibindo verbo “Amar” (Deusdado 2002)

Figura 12 - Gesto do número 0 no Gestuário 2D (Deusdado 2002)

Os resultados da avaliação ao sistema, identificaram que os surdos são muito exigentes em termos de usabilidade. O autor sacrificou o detalhe dos modelos gráficos por modelos leves de modo a obter um desempenho mais fluído. Pode-se apontar que esta decisão teve impacto nos resultados, tal como o próprio autor conclui, é necessário aplicações gráficas realistas de modo a serem aceites pelos utilizadores. Também foi verificado por Deusdado que os utilizadores preferem ver vídeos de pessoas em vez personagens.

Deste estudo posso conjecturar que uma aplicação destinada a surdos que use personagens virtuais, tem necessariamente de ter uma elevada qualidade gráfica nas animações, para essa aplicação estar à altura de uma que utilize vídeos com humanos, em vez de avatares.

Foi também interessante verificar que neste trabalho a própria comunidade surda identificou a necessidade de disseminação de dicionários de gestos que combatam o aparecimento de vários gestos para a mesma palavra.

2.2.3 Simon-the-Signer

Este trabalho e os das secções 2.2.4 e 2.2.5 foram subsidiados pela União Europeia como continuação do projeto ViSiCAST (Ralph, et al. 2000). Todos eles tiveram como objetivo melhorar o acesso à informação e serviços por parte dos cidadãos surdos recorrendo a um intérprete virtual. Foram utilizadas técnicas de captura de movimentos para reproduzir os gestos no avatar, sendo recolhidos por três canais diferentes (*Figura 14*):

- Luvas com 18 sensores resistivos para capturarem as posições dos dedos das mãos;
- Fato magnético para capturar as posições tridimensionais da mão, corpo e cabeça;
- Para as expressões faciais foi usada uma câmera com filtro de infravermelhos e 18 refletores posicionados na face do intérprete.

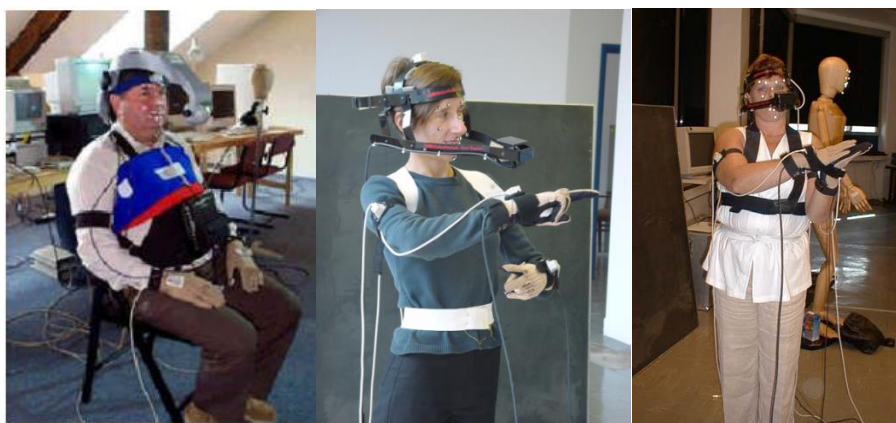


Figura 14 – Equipamento para a captura de movimentos a partir de um intérprete profissional de Língua Gestual. Utilizado nos trabalhos Simon-the-Signer, [secção 2.2.4](#), e [2.2.5](#), da esq. para a direita (Farzad, et al. 1999, Stephen Cox, et al. 2002, Verlinden, Tijsseling e Frowein 2002)

O Simon-the-Signer (Farzad, et al. 1999), também conhecido por SignAnim, é um avatar que traduz em tempo real, conteúdos noticiosos em *Sign Support English*, a partir de texto em Inglês (*Figura 15*).



Figura 15 - Intérprete virtual *Simon* (Farzad, et al. 1999)

A partir de uma base de dados de gestos, suportada pela técnica de captura de movimentos, o sistema estabelece a associação entre palavra e animação do gesto, conseguindo traduzir diretamente texto em língua gestual. Este processo de síntese do gesto é similar àquele que se pretende implementar.

A língua gestual escolhida no SignAnim foi o *Sign Support English (SSE)*. Ao contrário das línguas gestuais naturais, o *SEE* mantém a ordem das palavras da língua oral, facilitando o processo de tradução. No entanto, essa decisão teve implicações negativas na aceitação do sistema por parte do público-alvo, uma vez que a *SSE* não é adotada pela maioria da comunidade surda do Reino Unido. Por essa razão, os trabalhos desenvolvidos posteriormente (secções 2.2.4, 2.2.5 e 2.2.6), não utilizaram essa Língua Gestual.

É interessante notar que Farzad, et al. durante os testes, verifica que na ótica dos surdos a concatenação de vários vídeo-clips de gestos, não é uma solução prática, passando a citá-lo:

“Um grupo de teste da comunidade local de surdos, comentou que os “saltos” entre cliques de vídeo são intrusivos: consequentemente eles consideraram a leitura aborrecida e difícil.”

2.2.4 TESSA

Este trabalho (Stephen Cox, et al. 2002) tem como objetivo facilitar a comunicação entre surdos e funcionários, num posto de correio. Este sistema reconhece o discurso oral do funcionário, fazendo a sua tradução para gestos em Língua Gestual Britânica utilizando um avatar. O processo de síntese de gestos é similar ao do sistema anterior descrito (Simon-the-Signer). O sistema TESSA foi inicialmente desenvolvida pela UEA (*University of East Anglia*) em parceria com a empresa *TeleVirtual Ltd.* e com a colaboração do serviço de Correios britânico.

Uma das vantagens do uso de avatares em relação ao vídeo é a fácil manipulação do avatar e do ambiente 3D onde está inserido. O TESSA tira proveito dessas características, permitindo definir o melhor ângulo ou o nível de *zoom*, na apresentação do avatar, assim como também, alterar a sua identidade.

Durante a avaliação do sistema, o reconhecimento dos gestos foi razoável para um conjunto de frases, e bom, para um conjunto de palavras, contudo, o nível de satisfação foi medíocre. Com base nestes resultados, reforça-se a ideia de que além da elevada qualidade gráfica do avatar, é necessária uma animação facial e corporal credível, para que o sistema atinja níveis de satisfação elevados na ótica dos surdos.

2.2.5 Visia

O projeto Visia (Verlinden, Tijsseling e Frowein 2002) é uma aplicação *Web* que utiliza um intérprete virtual, para anunciar em Língua Gestual a previsão do estado do tempo. É semelhante ao projeto Simon-the-Signer, mas em vez de traduzir conteúdos noticiosos, traduz informação

meteorológica. Além disso, traduz o texto de várias línguas, designadamente, Inglês, Alemão ou Holandês, para a respetiva língua gestual natural (BSL, DGS e NGT). É necessária a instalação de um *plugin*, apenas disponível para o *browser Internet Explorer*, onde estão incluídos os ficheiros de movimento que representam os gestos, obtidos por captura de movimentos.

O Visia teve uma visão mais elevada no que toca à síntese do gesto utilizada em Simon-the-Signer e TESSA. Ao passo que os sistemas anteriores estabelecem a relação direta entre a palavra e a animação do gesto, Visia estabelece a relação entre a informação descritiva do gesto e a animação deste. Os autores criaram o SiGML (*Signing Gesture Markup Language*), um sistema que codifica os elementos que descrevem o gesto, designadamente a configuração da mão, da expressão facial, e configuração do movimento. O seu objetivo é codificar a informação descritiva do gesto, e não o seu significado, tornando-se um sistema independente da região da língua gestual.

O *interface* da aplicação Visia (*Figura 16*) pretende ser simples, no entanto, não contempla uma forma de manipular a velocidade da animação. Este é um fator importante para utilizadores que não se adaptem ao intérprete virtual e que tenham dificuldades em seguir a velocidade dos gestos.



Figura 16- Avatar Visia e *interface* da aplicação (Verlinden, Tijsseling e Frowein 2002)

2.2.6 eSign

A aplicação eSign (Zwitserlood, et al. 2005) tem como objetivo traduzir o conteúdo das páginas *Web* para língua gestual. O sistema é composto por duas componentes. A primeira consiste num avatar capaz de ser integrado em qualquer página *Web*, e que existe para traduzir qualquer conteúdo que o autor da página queira traduzir. A segunda componente consiste numa aplicação *Desktop*, que serve para criar o vocabulário em língua gestual que o avatar terá acesso. Esta aplicação é a continuação do projeto ViSiCAST, e usa muito da experiência adquirida nesse projeto.

A principal diferença para os projetos anteriores (2.2.3, 2.2.4, 2.2.5), é que o eSign não recorre ao processo de captura de movimentos para suportar a animação. O processo de animação consistiu em produzir em tempo real o movimento de um gesto estático para outro. Por um lado, este processo envolve algoritmos de *Inverse Kinematics* para definir os valores das rotações das articulações do modelo para cada postura. Por outro lado, envolve algoritmos de interpolação entre posturas, para definir a transição de uma pose para outra, de uma forma credível de acordo com a biomecânica do movimento humano (Kennaway 2002).

Analogamente ao Visia, o eSign utilizou SiGML para relacionar os gestos com a animação, que por sua vez, guiam o avatar com os movimentos necessários à composição do gesto. De igual modo ao sistema Visia, é necessário instalar um *plugin* para o *browser* de forma a visualizar o avatar a traduzir a informação do *Website*. O interface da aplicação (Figura 17) permite ao utilizador controlar certos aspetos do avatar como a posição da câmara virtual e parâmetros visuais como luz, entre outros.

A principal vantagem que eSign apresentou por não recorrer à técnica de captura de movimentos foi um custo menos elevado no processo de animação. Todavia, a grande desvantagem foi a elevada complexidade em desenvolver animações realistas. O grande avanço que este projeto deu ao estado-da-arte, foi a descoberta de um método que permite criar gestos de Língua Gestual sem a necessidade de produzir o gesto numa ferramenta de animação. O avatar aparenta estar melhor graficamente que os usados nos trabalhos anteriores, mas infelizmente não foram encontrados resultados da avaliação feita ao sistema. Seria interessante comparar os resultados de Visia que utiliza a captura de movimentos, com esta solução que não usa essa abordagem.

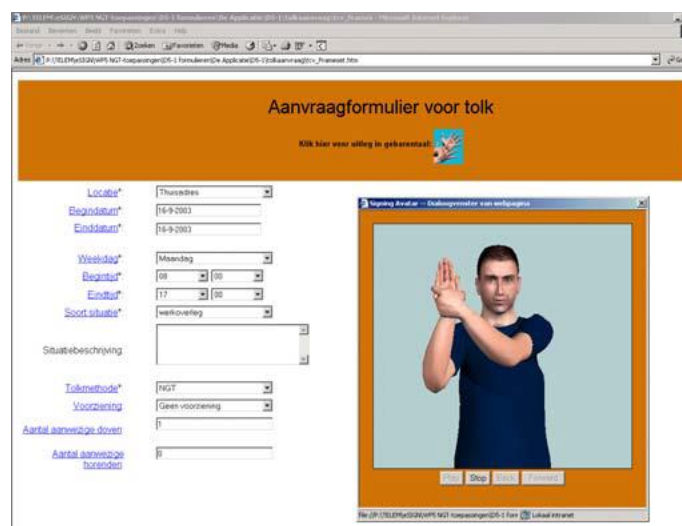


Figura 17 – eSign (Zwitserlood, et al. 2005)

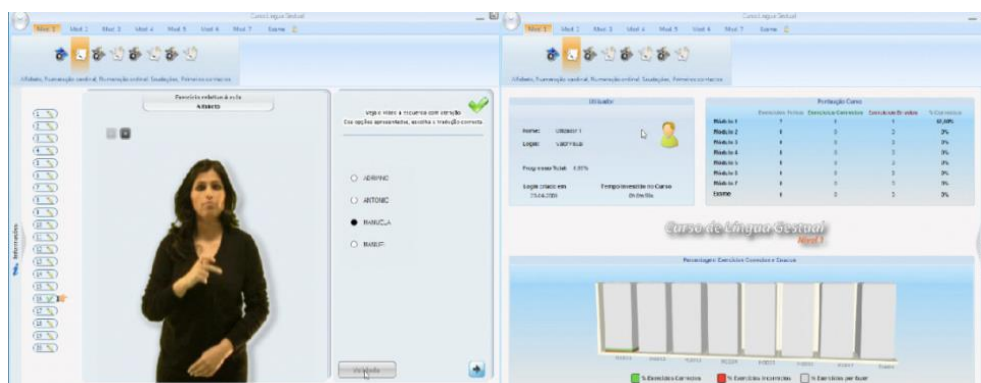
2.2.7 Curso de Língua Gestual em DVD

É uma aplicação multimédia interativa (ValorVisual 2009) destinada ao ensino da Língua Gestual Portuguesa. Os conteúdos programáticos do curso são apresentados em forma de vídeo e o curso é composto por 3 módulos, com níveis progressivos de dificuldade (*Figura 18*). No primeiro nível tem conteúdos como o alfabeto, numeração, cores, animais, etc. No segundo nível apresentam-se adjetivos, antónimos, verbos e pequenas frases. No terceiro e último nível, simulam-se conversas do quotidiano como diálogos em cafés, restaurantes, metro, bares, etc.



Figura 18 - Curso de Língua Gestual: conteúdos do primeiro ao terceiro nível (esq. para a direita)

O DVD da aplicação pode ser utilizado tanto na televisão, como no computador, contudo, no computador é possível além de visualizar as aulas, no final de cada lição, realizar exercícios para passar à lição seguinte. No fim de todos os exercícios, o utilizador tem acesso a um exame final. Ao longo do percurso didático é possível consultar a classificação, e após o exame final, imprimir o certificado. A *Figura 19* mostra um exemplo de exercício e a tabela de classificação.



da Vinci 2008). O dicionário está disponível também como aplicação para iPhone e Android com a mesma funcionalidade que o *Website*. Na *Figura 20* apresentam-se os *interfaces* para *Website* e Android:

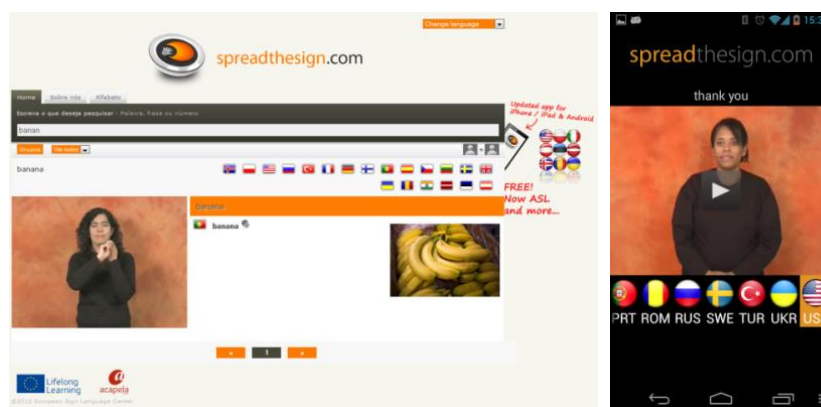


Figura 20 – Interfaces Web (esq.) e de Android (direita) do Spread the Sign

2.3.2 Signing Savvy

Signing Savvy (Signing Savvy LLC 2009) é um dicionário *online* para Língua Gestual Americana (ASL). Este dicionário além da consulta de gestos por categorias (animais, cores, números, etc.), também tem funcionalidades como: listas de palavras; construtor de frases, *quizzes* e *flash cards*. Estas funcionalidades tal como a versão para iPhone e Android da aplicação são funcionalidades pagas.

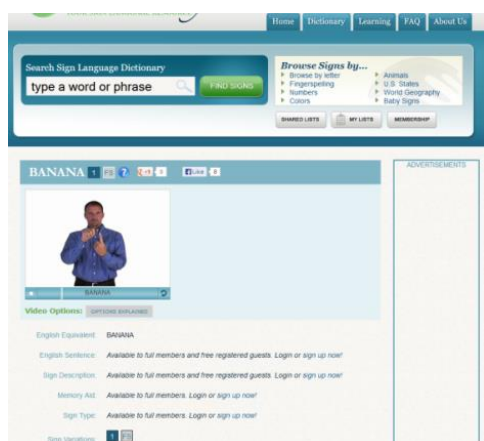


Figura 22 – Signing Savvy: consultar gesto



Figura 21 – Signing Savvy: construtor de frases

A *Figura 22* representa o *interface* que permite fazer a consulta de um gesto. Inclui a descrição da articulação, frases de exemplo em contexto, gestos alternativos e dicas de memorização. A *Figura 21* mostra o construtor de frases após a introdução de uma frase pelo utilizador. A frase introduzida é decomposta em várias palavras que podem ser alteradas ou ordenadas. O vídeo que

traduz a frase em ASL é composto pela concatenação de gestos de cada palavra na ordem da língua oral e não da língua gestual.

A funcionalidade listas de palavras serve para agrupar um determinado conjunto de conceitos numa lista. A partir desta, é possível fazer algumas coisas interessantes como consultar sequencialmente o seu vocabulário com vista à memorização, criar *flash cards*, ou *quizzes*. Um *quiz* corresponde a um conjunto de perguntas aleatórias, alternando entre dois tipos de questões. O utilizador é convidado a adivinhar dado um vídeo de um gesto, qual é a sua palavra correspondente, ou o processo inverso, dada uma palavra o utilizador tem de adivinhar qual o gesto correto. Um *flash card* é essencialmente conjunto de gestos que o utilizador visualiza um gesto, apreende-o, e depois visualiza a palavra respetiva. Em seguida, passa ao próximo gesto e repete o processo. É uma ferramenta que serve de memorização e permite preparar o utilizador para os *quizzes*.

2.3.3 Dicionários Gestuais da VCom3D

A VCom3D lançou uma família de 7 dicionários (VCom3D s.d.). Os temas estão distribuídos entre Ciência, Matemática, Terra, Ciência da Vida e Física. Estes são dicionários ilustrados direcionado para crianças surdas ou com deficiências auditivas, e que usem a Língua Gestual Americana ou *Signed English*. Cada dicionário (*Figura 23*) permite consultar palavras específicas ao universo do seu tema particular. Além de cada palavra que consta no dicionário, a maioria do texto do *Website* é passível de ser traduzido pelo intérprete virtual, um avatar. Para se conseguir visualizar o avatar é preciso instalar um *plugin* disponível apenas para o *browser Internet Explorer*. Todos os dicionários apresentam as palavras por ordem alfabética e em particular dois, permitem visualizar as palavras também por categoria (i.e. animais, flores, frutas, etc.). Incluem o tipo de palavra (i.e. verbo, substantivo, adjetivo), plural, descrição do significado, palavras relacionadas, e uma imagem representativa do conceito. O dicionário permite escolher para intérprete, um avatar de entre várias etnias ou gêneros, e permite rodar a sua posição para visualizar o gesto em perspetivas diferentes. A velocidade da animação também pode ser definida para mais lenta ou mais rápida como preferência.

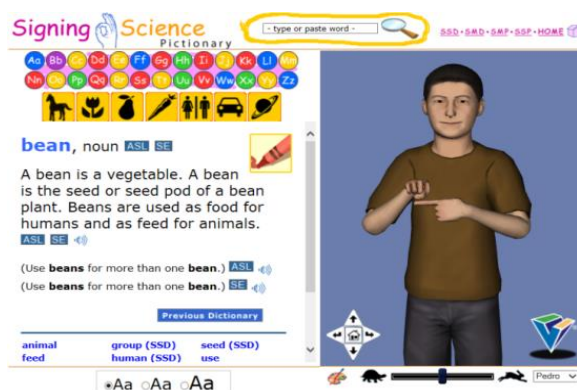


Figura 23 – Signing Science Pictionary

2.3.4 Análise comparativa entre Dicionários Gestuais

A Tabela 2 sintetiza os pontos mais importantes de cada uma das aplicações do tipo dicionário. São comparadas quatro características: composição de frases, componente para testar os conhecimentos, uso de LGP, existência de versão *mobile*, e uso de Avatar.

Plataforma	Composição de Frases	Componente para testar os conhecimentos	LGP	Versão <i>mobile</i>	Avatar
Spread the Sign			X	X	
Signing Savvy	X*	X		X	
Dicionários da VCom3D				X	X

Tabela 2 – Propriedades principais nos Dicionários Gestuais descritos

Constata-se que apenas o Spread the Sign tem tradução para LGP, contudo, de entre todos os sistemas, é o menos rico em funcionalidades. No outro extremo temos o Signing Savvy como o mais completo, apresenta composição de frases, e componente para testar os conhecimentos, nomeadamente através de *Quizzes* e *Flash-Cards*. É de notar que a composição de frases é feita pela concatenação de vídeos, não sendo o método mais adequado. É também de realçar que neste sistema a ordem dos elementos da frase não corresponde à ordem de elementos da língua gestual. Os dicionários da VCom3D são os únicos que usam como intérprete um Avatar, porém, este apenas se focam na componente de consulta e tradução de palavras. Além disso, tiram partido das vantagens de uso de avatares como: a possibilidade de escolha de um intérprete de uma lista de avatares e a manipulação da posição da câmara para mostrar o gesto em ângulos diferentes.

2.4 Sumário

Neste capítulo, deu-se uma visão geral sobre os conceitos essenciais ao entendimento dos temas abordados. Destacaram-se os projetos relacionados com os temas desta tese, designadamente, a Língua Gestual e Avatares. Por fim, identificaram-se alguns Dicionários Gestuais, analisando-se as suas características mais relevantes.

* Composição de frases através da concatenação de vídeo não é a mais indicada para surdos em (Farzad, et al. 1999).

Capítulo 3

Investigação e Testes Prévios

No geral, este trabalho enquadra tecnologias relativamente recentes, como o sensor Microsoft Kinect[®], por sua vez, são abordados desafios ainda não triviais, especialmente neste capítulo, como o *finger tracking*. Estes motivos conduziram a uma abordagem de trabalho exploratória, levando ao desenvolvimento de algoritmos, aplicações, e à redefinição de metas ao longo do decorrer da tese.

Este capítulo, inicia-se pela apresentação dos motivos da escolha do *MotionBuilder* como *software* de animação. Em seguida, apresenta e compara dois tipos de animação: a animação pelo método manual, que resultou num avatar que soletra palavras em LGP; e na animação assistida pela captura de movimentos, focada para a interpretação de palavras de LGP.

O capítulo é concluído, com a investigação de um método de *finger tracking* que complementa a captura de movimentos do corpo obtida pelo Kinect, com a posição tridimensional dos dedos ao longo do movimento. Esta abordagem resultou na investigação de um método de *finger tracking* semiautomático, que envolve a ajuda do utilizador para detetar as articulações dos dedos.

3.1 Escolha do *software* de animação 3D

Existem várias ferramentas de *software* para animação 3D como: *Maya* (Autodesk 2011), *Blender* (Blender Foundation 2011), *Animation:Master* (Hash, Inc 2011), *Poser* (Smith Micro Software, Inc 2011), *MotionBuilder* (Autodesk 2011), entre outras.

O *MotionBuilder* foi escolhido por ser um *software* muito eficiente no suporte à animação por captura de movimentos, com o qual já estava familiarizado. Outra vantagem relevante para a sua escolha, foi o facto de este disponibilizar um *SDK* para desenvolver aplicações que tiram partido das suas funcionalidades, em linguagens como C++ e Python.

3.2 Adaptação e primeiros testes em animação 3D

Este teste foi importante porque trabalhei a importação de um avatar e sua integração no *software*, e por outro lado, modeleí gestos estáticos (i.e. gestos de letras) e respetiva animação e comprovei a capacidade de interpolação do *MotionBuilder*.

O primeiro passo consistiu em procurar modelos gratuitos de humanos virtuais⁶ a fim de escolher um para a realização de testes. O modelo que escolhi (*Figura 24*) já tinha definido a estrutura de ossos e articulações, o que facilitou a manipulação do modelo. O passo seguinte consistiu em modelar dois gestos distintos e criar uma animação que fizesse a transição de um gesto para outro. Interessava-me fazer o soletrar palavras, porque em LGP este processo é usado para descrever palavras que não tenham a representação do gesto em LGP de uma palavra em LP, ou então, quando se pretende soletrar nomes próprios ou siglas.

Uma vez que a cada articulação do esqueleto, não está convencionado um identificador único, não é possível ao *MotionBuilder* determinar automaticamente se a articulação “x” é a articulação da cabeça ou dos pés. Deste modo o passo seguinte à importação do modelo foi proceder à sua *Characterization*. Este processo consiste em mapear o nome de cada articulação do modelo com o nome padrão que o *MotionBuilder* identifica para essa articulação.

Seguidamente procedi à produção da animação: usando o sistema de animação por *keyframe*, criei duas *keyframes*, o modelo na primeira executava em LGP o gesto para a letra “A”, e nalgumas *frames* a seguir executava o gesto para a letra “O”.



Figura 24 – Gestos estáticos para as letras “A” (esq.) e “O” (direita) em LGP no *MotionBuilder*

A transição obtida na animação entre o gesto de uma letra e o gesto de outra revelou-se fluída. No seguimento deste resultado, decidi criar uma aplicação para traduzir qualquer letra em língua gestual portuguesa, importante como etapa preliminar para chegar à criação de gestos de palavras, e por conseguinte, construção de frases obtida pela combinação de palavras. O desafio principal foi construir a animação dinamicamente com a transição de gestos para letras arbitrárias. Para isso, aprendi os conceitos básicos da linguagem de *scripting* Python, antes do desenvolvimento.

⁶ <http://www.turbosquid.com> – acedido a 22 de Setembro de 2012

3.3 Soletrar palavras em LGP – criação de *plugin*

Esta aplicação gera uma animação em tempo real de um avatar a traduzir para gestos de LGP cada letra introduzida pelo utilizador e foi desenvolvida em Python como um *plugin*⁷ do *MotionBuilder*.

Em primeiro lugar criei os gestos de todas as letras de A a Z no *MotionBuilder*. Modelei cada gesto definindo a configuração dos braços, mãos e dedos. Cada gesto foi armazenado como uma *Pose*⁸. Além disso, criei uma pose neutra usada para a posição inicial do avatar durante a animação. O avatar utilizado foi um dos modelos existentes no *MotionBuilder* designado por Mia.

Após a criação da biblioteca de gestos ou poses, foi necessário aceder a cada uma, através da linguagem de *scripting* Python e construir dinamicamente uma animação. Para tal, defini uma *keyframe* para cada letra, com um intervalo de 5 *frames*. Ao executar a animação, o *software* faz automaticamente a passagem da pose de uma *frame* para a pose da *frame* seguinte. O valor de 5 *frames* para o intervalo entre as letras, foi obtido empiricamente como um ótimo valor para uma animação fluída.

Foi desenvolvido um *interface* simples (*Figura 25*) que inclui uma caixa de texto para o utilizador escrever uma palavra, e um botão para gerar a animação a partir do texto que o utilizador pretende ver soletrado.

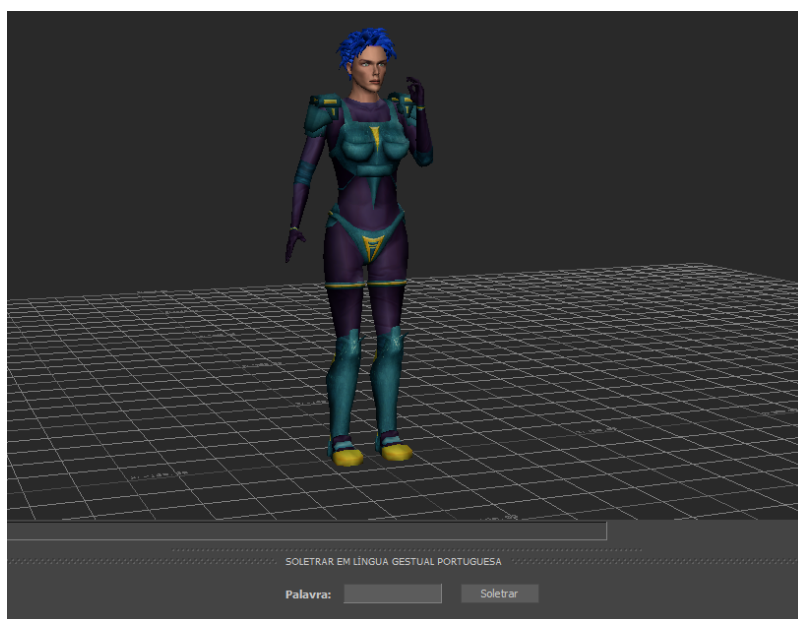


Figura 25 – *Plugin* Soletrar palavras em LGP, situado na margem inferior do *MotionBuilder*

⁷ O conceito de *plugin* também é conhecido no contexto do *MotionBuilder* como *Tool*. Consiste em várias instruções de código escritas usualmente em Python para desempenhar uma ou várias tarefas de forma automática.

⁸ Pose no contexto do *MotionBuilder* determinada a postura do avatar, através da representação a posição e orientação de todas as suas articulações.

Concluí ao longo do desenvolvimento desta solução que modelar um gesto para cada letra do alfabeto é uma tarefa muito demorada. Por conseguinte, modelar algo mais sofisticado como palavras seria ainda mais trabalhoso, pelo que faz sentido recorrer a um sistema que torne o processo de produção de gestos mais eficiente. O caminho seguinte foi usar um sensor de movimentos para capturar o movimento apenas corporal, mas desta vez pretendo animar gestos de palavras em vez de letras.

3.4 Animação de gestos assistida por Kinect

Antes de proceder à captura, pesquisei qual o sensor de profundidade mais adequado ao objetivo. Dessa pesquisa resultaram os sensores mais recentes e de baixo custo como o Microsoft Kinect® e o Leap Motion®. A comparação de ambos que levou à escolha de um deles está definida na secção dos *Sensores de profundidade* do Capítulo 2.

Primeiro teste ao *motion capture*

Após a escolha do sensor, foi necessário fazer a ponte entre o Microsoft Kinect® e o *software* de animação. Fiz uma pesquisa pelo que já existia sobre a integração dos dados do Kinect em plataformas de animação, e encontrei algumas ferramentas que criavam esta ponte. Tais como: *iPi Recorder* (iPi Soft 2011), *iClone 5* (Reallusion 2011), *Miku Miku dance* (Higuchi 2008), *Blender Motion Capture Addon* (NI MATE 2011), *Easy Mocap* (Easy Mocap 2011) e *Brekel Kinect* (Brekelmans 2011). Optei pelo *plugin* para o *MotionBuilder* chamado *Brekel Kinect*, porque além de ter um bom desempenho na passagem do movimento humano para o avatar, já estava integrado com o *software* de animação que estava a usar. O próximo passo foi proceder a um teste de captura de movimentos. Depois de aberto o *software Brekel Kinect* (Figura 26), é preciso garantir que está à distância adequada para detetar o esqueleto completo do utilizador. Em seguida, liga-se a receção dos dados no *MotionBuilder* (Figura 27), e procede-se à gravação do movimento num esqueleto predefinido pelo *plugin*. A informação transmitida consiste nas posições tridimensionais de cada articulação detetada pelo sensor. Por fim, é necessário associar o movimento do esqueleto ao avatar.

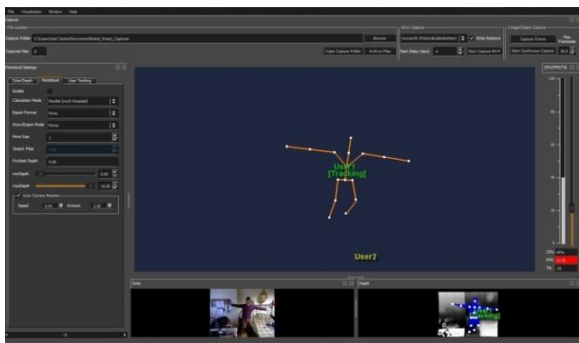


Figura 26 – *Brekel Kinect* (captura)

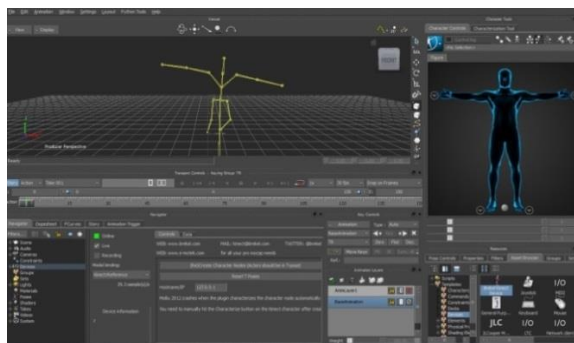


Figura 27 – *MotionBuilder* (captura)

O resultado foi a captura de movimentos do tronco humano, e transpostos para uma personagem definida no *MotionBuilder*. Esta captura foi gravada no formato de ficheiro proprietário da aplicação (FBX), podendo ser reutilizável para animar outra personagem. Na *Figura 28*, vê-se o esqueleto de cor amarela (*Stick Figure*), representando as articulações do utilizador detetadas pelo sensor, e do seu lado direito, a personagem Mia conduzida pelo movimento do esqueleto.

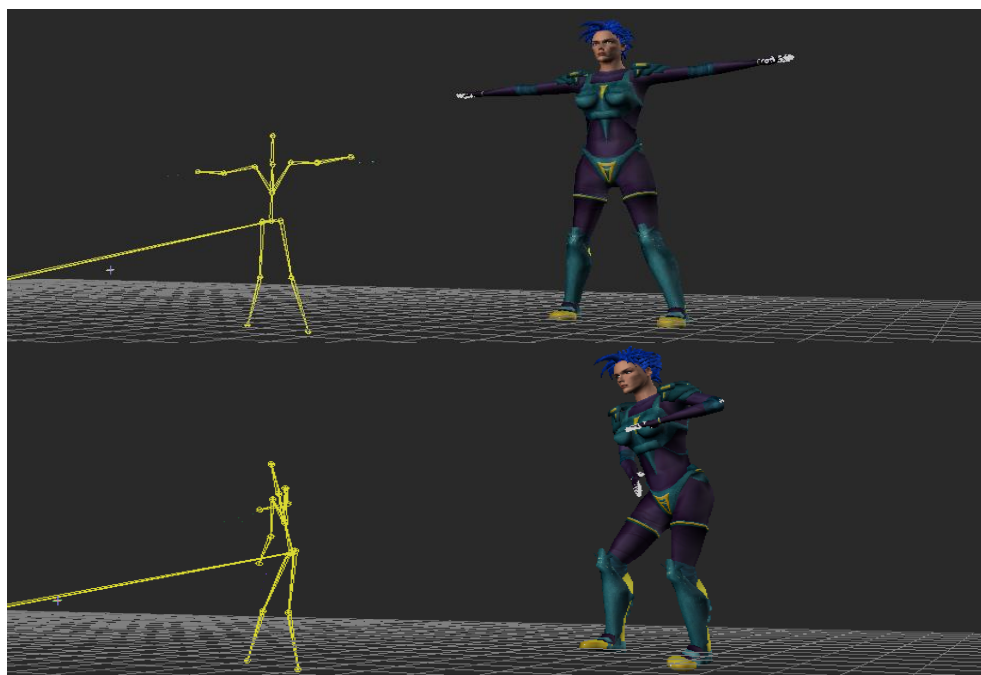


Figura 28 - Captura de movimentos integrada num avatar em *MotionBuilder*

Animação de gestos de palavras de LGP

O passo seguinte após a animação teste, consistiu em gravar gestos de palavras e fazer uma comparação destes, com os mesmos gestos porém, realizados através de animação manual por *keyframe*, confrontando o tempo de produção e a qualidade de animação de cada abordagem. Foram escolhidas 3 expressões para a captura: a palavra “*Olá*”, o soletrar da expressão coloquial “*LOL*”, e a expressão “*Ondas do mar*”.

Ao contrário da animação criada manualmente por *keyframes* (3.2) a animação assistida pelo Kinect necessita de um processo de produção mais complexo. Este processo foi identificado como o seguinte *pipeline*:

1. Captura do movimento
2. Ligação do avatar aos dados do Kinect
3. Tratamento de dados
4. Adição da animação da mão e dedos à animação corporal do avatar

O primeiro passo deste processo é gravar o movimento do gesto através do *plugin* que envia a informação do Kinect para o *MotionBuilder*.

A tarefa seguinte consiste em fazer a ligação dos dados capturados ao avatar que se quer animar. Para isso faz-se a *Characterization*⁹ do *Stick Figure* (esqueleto controlado pelos dados do Kinect) e posteriormente define-se esse personagem como *Input Source* no avatar.

Após a ligação do avatar aos dados vindos do Kinect, procede-se ao *Data Cleaning*, isto é, ao processo de tratamento de dados. Este passo existe, porque os dados capturados 99% não podem ser integrados diretamente no avatar. Por este motivo, o *Data Cleaning* tem dois objetivos. O primeiro é determinar a posição correta de cada articulação no conjunto de instantes da captura. Isto é necessário quando o sensor não deteta com exatidão a posição do osso durante o movimento (*Figura 29*), ou no caso extremo da articulação não estar a ser temporariamente detetada. O segundo objetivo é corrigir aquelas posições que aparentam estar bem no esqueleto do *Stick Figure*, mas ao passarem para o esqueleto do avatar não têm a mesma correspondência. Isto acontece porque a fisionomia do esqueleto do *Stick Figure*, apesar de próxima, não é exatamente igual à fisionomia do esqueleto do avatar. Este tipo de falha inclui as posições quando o movimento real faz o braço ou a mão passar muito junto ao corpo, e no avatar essa extremidade trespassa o corpo do personagem (*Figura 30*).

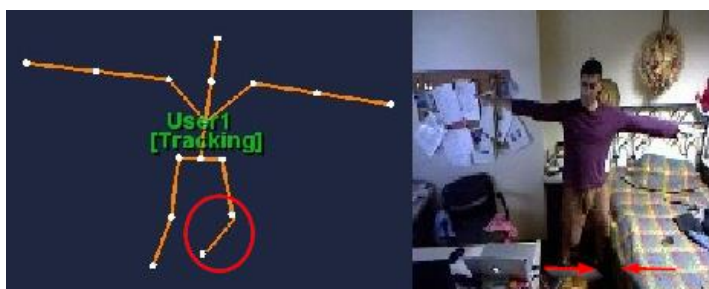


Figura 29 – Falha no *tracking*



Figura 30 – Braço trespassa corpo do avatar

O último passo é o chamado *hand-over* manual. Este passo consiste em fazer a animação manual da mão por cima da animação capturada com o Kinect. Isto resulta em duas animações sobrepostas, ou seja, por um lado temos a animação do corpo capturada com o Kinect, e por outro, uma animação da mão feita manualmente por *keyframes*. Este passo usa intensamente a animação manual, pois é necessário definir a configuração e orientação da mão num intervalo temporal. É preciso ter um bom entendimento da descrição do gesto, perceber as pequenas *nuances* do movimento, de modo a se atingir a definição da pose correta no instante correto. Especialmente neste passo, fiz pequenos ajustes tendo em conta o gesto no todo, e não apenas na soma das suas partes. Quer isto dizer, por exemplo, que por vezes é necessário aumentar a velocidade do

⁹ Processo descrito em *Adaptação e primeiros testes em animação 3D*.

movimento, por vezes diminuir, ou realçar outros pormenores de modo a produzir um gesto fidedigno. Sem alterar muito o original, se não corre-se o risco de ficar pouco fluído e “robótico”.

Comparação entre animação manual e animação capturada

A animação feita manualmente resultou numa animação menos fluída, em relação à realizada com a ajuda do Kinect. Esse facto é visível na *Figura 31*, onde se vê do lado esquerdo o avatar com uma postura descontraída enquanto no lado direito, está bastante mais rígida. O tempo de produção de todos os gestos foi na média o mesmo, para as duas abordagens.

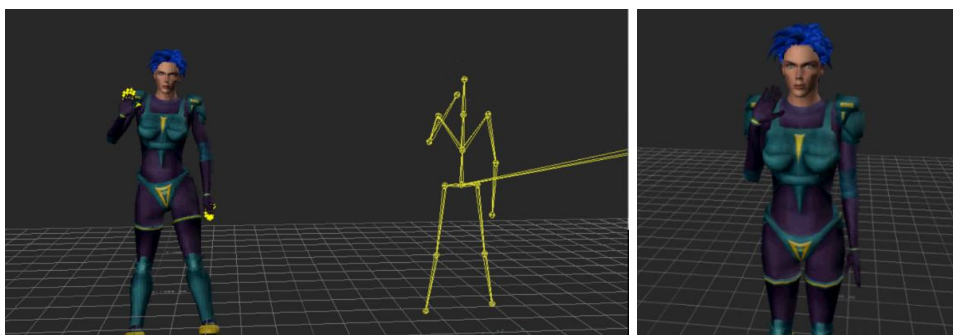


Figura 31 - Comparação entre animação assistida por Kinect (esq.) e animação manual (dir.)

O uso de animações assistidas por Kinect tem a grande vantagem de obter resultados mais realistas que recorrendo a animações manuais. Por um lado, exige-se algum trabalho, designadamente com o tratamento de dados de forma a corrigir falhas na captura; por outro lado, é compensatório, dado o resultado realista da animação.

Apesar da animação obtida por edição manual, esse fator está dependente do nível de experiência do animador e da complexidade do gesto. Todavia, este tipo de animação pouco natural é difícil, dada a complexidade da biomecânica humana. O recurso ao Kinect é também vantajoso, porque diminui o tempo necessário na criação de animações caso o gesto não tenha muitas falhas nas posições durante a captura. No caso do gesto para a expressão “*Ondas do Mar*”, este tinha muitas falhas e fez com que na média o tempo de produção fosse igual para as duas abordagens. Deste teste concluiu-se que é possível fazer uma captura de movimentos com o sensor Kinect e animar um avatar com esse movimento. No entanto, o sensor tem *tracking* nativo do movimento do corpo e da face, mas não dos dedos. No âmbito da língua gestual, a configuração da mão é uma componente essencial na descrição do gesto. O passo seguinte envolveu investigar quais os dados necessários à transposição do movimento dos dedos, para um modelo da mão tridimensional.

3.5 Definição de gesto por coordenadas – criação de *plugin*

O próximo *plugin* desenvolvido para o *MotionBuilder* tem como objetivo testar a informação necessária para determinar a configuração da mão num avatar, e foi desenhado tendo em conta três requisitos: carregar uma postura num avatar a partir de um ficheiro, salvar para ficheiro a pose do avatar ativo, e produzir uma animação a partir da leitura de vários ficheiros. Antes de iniciar o seu desenvolvimento, fiz uma análise à estrutura de ficheiros com extensão BVH¹⁰. Verifiquei que a informação essencial para definir uma postura é representada para cada articulação pelas coordenadas de translação e rotação em torno dos três eixos principais (i.e. x, y e z).

Em seguida apresento os *interfaces* da aplicação a que chamei *Poses Animator*:

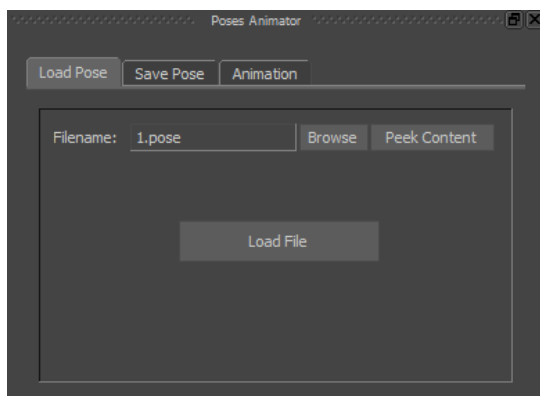


Figura 32 - *Poses Animator*: interface para carregar no avatar gestos estáticos

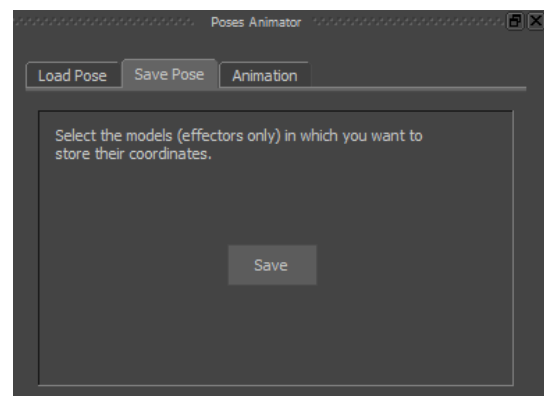


Figura 33 - *Poses Animator*: interface para salvar em ficheiro gestos estáticos

O *interface* da *Figura 32* permite carregar ficheiros de gestos, para isso basta carregar no botão *Browse*, escolher o ficheiro, e por fim carregar na opção *Load File*. O gesto contido no ficheiro irá ser carregado para o avatar que está a ser utilizado como ativo no *MotionBuilder*. Tecnicamente, esta operação lê as coordenadas de translação e rotação, e define os seus valores para cada articulação do avatar. É possível ver o conteúdo de cada ficheiro, carregando na opção *Peek Content* como se pode ver na *Figura 34* **Erro! A origem da referência não foi encontrada..**

O *interface* da *Figura 33* permite armazenar gestos estáticos (i.e. gestos de letras) em ficheiro, salvando as coordenadas de todas as articulações do avatar. É possível escolher as articulações que se pretender armazenar, selecionando-as primeiro no personagem, e em seguida clica-se na opção *Save*. Daí aparece a opção para escolher a pasta e o nome do ficheiro que se pretende salvar.

¹⁰ *.bvh é um formato de dados próprio para captura de movimentos, pois armazena informação hierárquica dos ossos de um esqueleto tal como informação do seu movimento. Um *software* de animação que trabalhe com este formato, consegue mapear a informação do esqueleto e do seu movimento, e associá-los a um esqueleto de outro personagem.

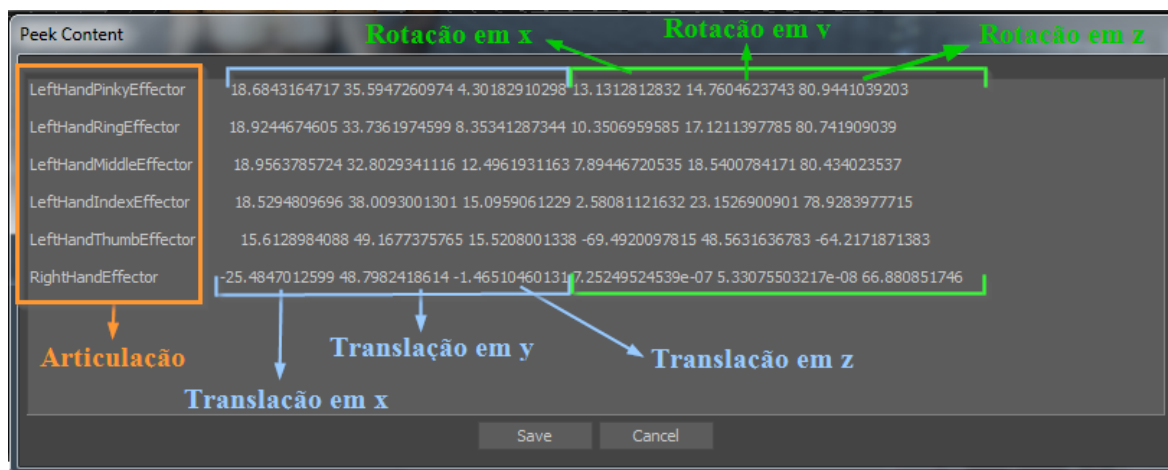


Figura 34 - Poses Animator: interface para visualizar e editar o conteúdo do ficheiro de gestos

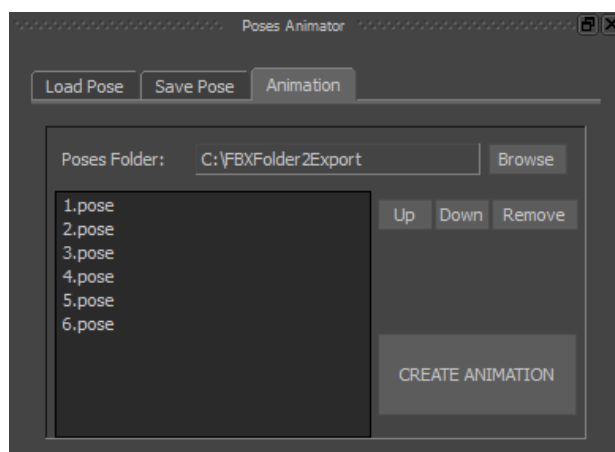


Figura 35 - Poses Animator: interface para gerar a animação

A interface da Figura 35 permite gerar animação com gestos arbitrários que foram armazenados em ficheiros. Esta interface também permite ao utilizador pré-visualizar o gesto contido num determinado ficheiro. Ao se carregar em cima deste duas vezes, por exemplo em “1.pose” (Figura 35), o gesto para representar o número “1” em LGP é carregado no avatar ativo do *MotionBuilder*. O sistema de transição de gestos estáticos do *Poses Animator*, consiste em distribuir várias *keyframes* por diferentes instantes temporais, onde em cada um se representa um gesto diferente. Este método baseia-se no *plugin* Soletrar palavras já descrito na 3.3

Este desenvolvimento, testou que as coordenadas de translação e rotação são suficientes para definir dinamicamente a configuração da mão num gesto de LGP. No fim, obtive uma solução com interesse particular para armazenar e representar gestos de letras num avatar, mas que pode ser usado para lidar com qualquer postura estática que se queira representar. O próximo passo foi capturar as coordenadas da mão do utilizador, e por conseguinte, animar um avatar.

3.6 *Hand tracking – desenvolvimento de algoritmos*

Para detetar as mãos foram desenvolvidos dois algoritmos com diferentes abordagens, utilizando o sensor Microsoft Kinect[®]. Escolhi a linguagem C# para o desenvolvimento por ser uma linguagem de alto nível e com a qual me sinto confortável.

A **1ª abordagem** baseou-se no princípio geral que as mãos são as extremidades mais próximas do sensor durante a execução de um gesto. O método *Depth Segmentation* consiste em analisar todos os pontos dentro de um intervalo de profundidades, e descartar os restantes. Considerando que as mãos do utilizador vão estar mais próximas do sensor que as restantes partes do corpo, admite-se que as mãos vão se encontrar em dois limites de profundidade distintos, um limite máximo e um mínimo. No entanto, *Depth Segmentation* por si só, não cobre os casos em que o utilizador se afaste/aproxime do sensor, ou, quando o sensor é posicionado a uma distância diferente do valor preestabelecido, saindo fora do intervalo. Para contornar estes problemas, decidi calcular o intervalo de profundidade em cada *frame*, assim sendo, o intervalo adapta-se às circunstâncias descritas.

Para calcular a amplitude do intervalo de profundidade usei a seguinte fórmula: defini para o limite inferior, a distância mínima a que o utilizador se encontra do sensor; e para o limite superior usei o valor do limite inferior mais 150mm. O valor 150mm foi testado várias vezes, e cheguei à conclusão que era um valor suficientemente abrangente para definir a distância entre o centro do pulso até à ponta do dedo. Para filtrar do intervalo todos os pixels que pertençam ao ambiente em volta da mão, verifico se cada pixel corresponde ao conjunto de pontos que o Kinect deteta serem do utilizador. Caso existe essa correspondência, então considero que o ponto do intervalo de profundidade faz parte da mão. Na *Figura 36*, pinte de cor amarelo todos os pixels que pertençam ao utilizador, enquanto a azul pinte os pixels respetivos às mãos.

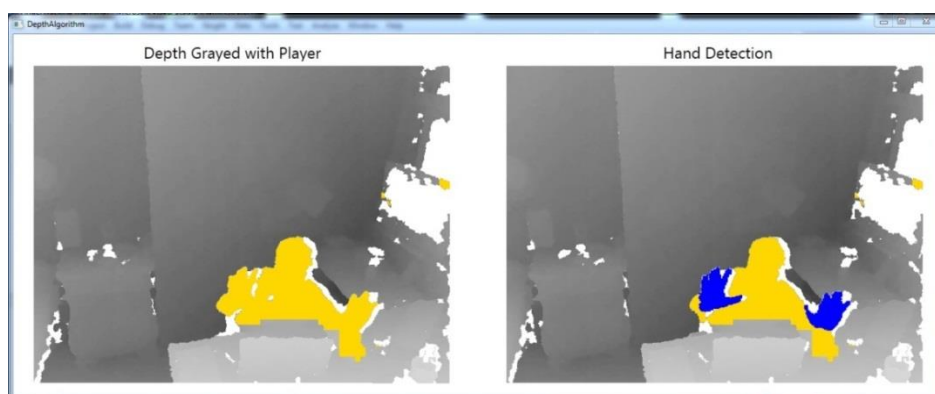


Figura 36 - Algoritmo de *hand tracking* por *Depth Segmentation*

A **2ª abordagem** é mais simples que a primeira, pois o Kinect fornece as coordenadas para qualquer articulação do corpo do utilizador. Sabendo essa coordenada, o algoritmo desenha uma elipse à volta da articulação de cada mão. Ambas as imagens da *Figura 37* revelam os resultados

do mesmo algoritmo, contudo apresentam-nos de maneira diferente. A imagem da esquerda mostra elipses à volta das mãos enquanto a imagem da direita define pontos brancos para o centro de cada mão. Criaram-se duas vistas diferentes do algoritmo para se verificar com exatidão o centro da mão.

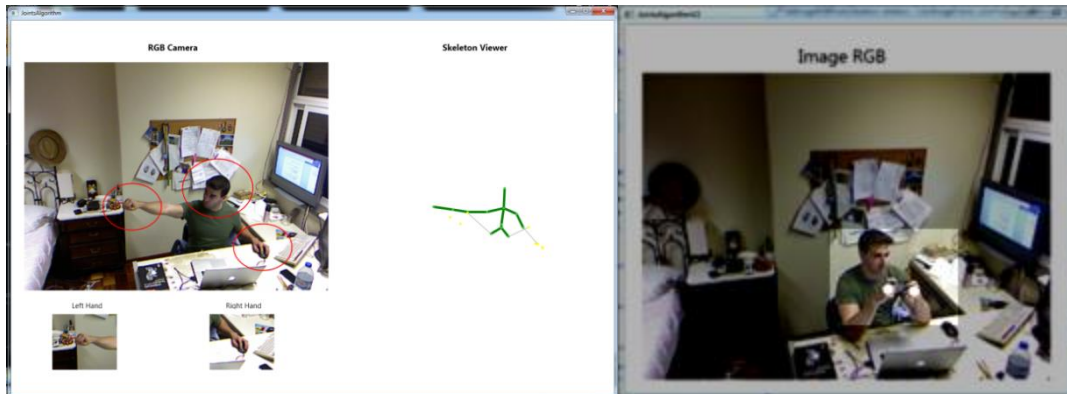


Figura 37 - Algoritmo de *hand tracking* usando as articulações detetadas pelo Kinect

Na *Figura 37*, especificamente na imagem do lado esquerdo, descreve-se para efeitos de teste uma elipse para a articulação da cabeça, um esqueleto com os ossos e articulações representativos do utilizador (boneco verde), e as imagens da mão esquerda e direita isoladas do resto do corpo. Estas últimas podem servir como base para detetar os dedos, bastando excluir os píxeis que se encontram numa profundidade diferente da profundidade da mão.

Após o desenvolvimento destes dois algoritmos, criei uma aplicação teste que executa os dois algoritmos simultaneamente (*Figura 38*) para efeitos comparativos.

Comparação das duas abordagens

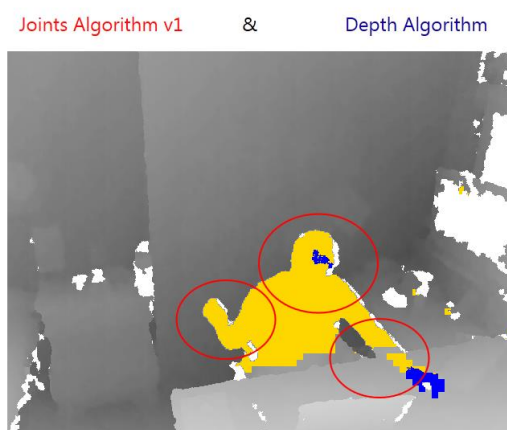


Figura 38 - Comparação entre os algoritmos de *hand tracking* desenvolvidos

Na *Figura 38*, mostra-se os resultados das duas abordagens. Neste caso particular, nota-se que não tiveram totalmente sucesso ambos os métodos. Em relação à mão direita, o algoritmo da 2ª

abordagem detecta a mão, enquanto o algoritmo por *Depth Segmentation* não (1ª abordagem). Relativamente à mão esquerda nota-se que a 1ª abordagem é mais eficaz, embora a 2ª abordagem também detecte a mão, não apresenta com precisão o centro da mão¹¹. A bola vermelha na cabeça é só para efeitos de teste no desenvolvimento e não tem nenhuma implicação no resultado da deteção das mãos. Já o algoritmo por *Depth Segmentation* identifica uma mancha azul na cabeça, significando que está a indicar a cabeça como mão. Este é um dos problemas da abordagem baseada em intervalos de profundidade, pois se algum ponto preencher as condições que o fazem pertencer ao intervalo, ou seja, a mão chegou muito perto da cabeça, o algoritmo identifica-o como sendo um ponto da mão. Um possível melhoramento para esta situação, seria ajustar o limite superior e diminuir o valor de 150mm para um valor inferior, de modo a restringir um pouco mais o intervalo de profundidade.

Durante todos os testes, o algoritmo da 2ª abordagem, o algoritmo que usa as articulações do utilizador fornecidas pelo Kinect, foi mais preciso e eficaz que o anterior. É razoável conjecturar que pode ser melhorado, através da mistura de características do primeiro algoritmo, e assim gerar um algoritmo mais robusto que o atual. Porém, o algoritmo reúne condições suficientes para se passar ao *finger tracking*, ou seja, à deteção e localização dos dedos.

3.7 *Finger tracking* - testes

O *finger tracking* não é uma tarefa trivial e é um dos maiores desafios encontrados nesta tese. Existem avanços nesta área utilizando o Kinect, nomeadamente por Stegmüller através do seu contributo com a biblioteca *Candescent NUI* (Stegmüller 2012). Para ter uma ideia da complexidade da deteção dos dedos, realizei alguns testes com esta biblioteca.

O teste a esta biblioteca teve como objetivo testar gestos simples, as letras do alfabeto de LGP. Para isso, verifiquei quais as letras mais difíceis de detetar, e as que não traziam problemas. Além disso, testei o algoritmo de *tracking* em duas perspetivas diferentes.

Os pontos vermelhos na imagem inferior da *Figura 39* representam as pontas dos dedos detetadas pelo algoritmo. Em ambas as perspetivas foi detetado a ponta do dedo do meio, porém, não é claro o contorno do resto dos dedos. O teste decorreu para todas as letras do abecedário e os resultados foram similares. O algoritmo em média detetou um, dois dedos por gesto. Todos os gestos têm dedos que ficam ocultos, por vezes tapados por outros, noutras vezes porque não estão de frente para o sensor.

¹¹ As elipses são uma melhor forma de apresentar o resultado da 2ª abordagem, por isso neste teste deixou-se a vista dos pontos brancos de fora.

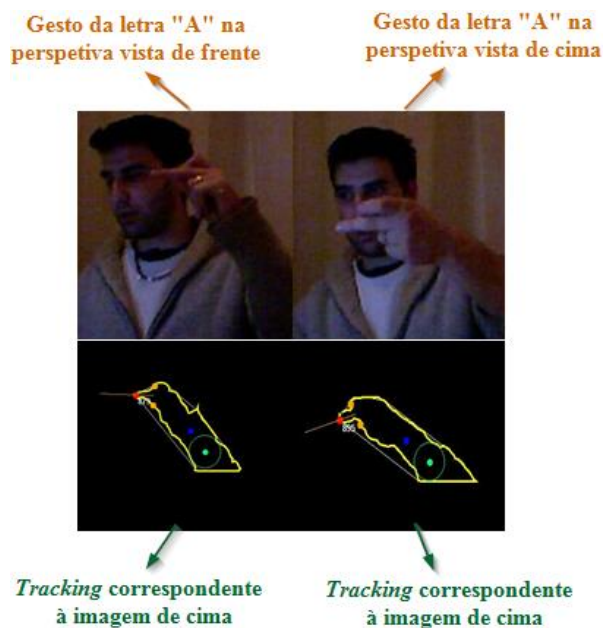


Figura 39 – Gesto da letra “A” em LGP com duas perspectivas diferentes

O *finger tracking* aplicado à língua gestual é muito mais complexo do que apenas detetar as pontas dos dedos, derivado das oclusões dos dedos durante a execução de um gesto e da falta de precisão nos dados de profundidade relativos às extremidades da mão. À luz destes resultados, decidi investigar um pouco mais longe e desenvolver alguns testes com uma biblioteca de processamento de imagem desenvolvida em C#.

AForge.NET

AForge.NET é uma *framework* de processamento de imagem e inteligência artificial. A abordagem do teste anterior realizada por Stegmueller, utilizava os dados de profundidade do Kinect. Uma vez que esses dados ao nível dos dedos não são muito precisos, a minha abordagem passou por tentar obter o contorno dos dedos através da imagem de cor RGB.

O primeiro passo consistiu em retirar os pixéis que não correspondiam aos pontos da mão ou braço. Então apliquei um filtro de profundidade para tudo que estivesse a 70mm de distância da articulação da mão, ficaria a preto. O valor de 70mm foi um valor que restringiu mais o intervalo de profundidade usado anteriormente (secção 3.6), mas empiricamente gerou melhores resultados pois geralmente detetava a mão e o braço como pretendia.

Para detetar os contornos foi aplicado um algoritmo chamado *Canny Edge Detector*. Este algoritmo pinta a branco os contornos dos objetos de uma imagem de entrada e tudo o que não seja contorno, pinta a preto. Para destacar mais os contornos, adicionei um pouco de claridade à imagem de entrada do *Canny*, imagem que tem de estar em tom de cinza. Os resultados do algoritmo podem ser vistos nas Figuras 40, 41 e 42 para vários gestos de letras de LGP.

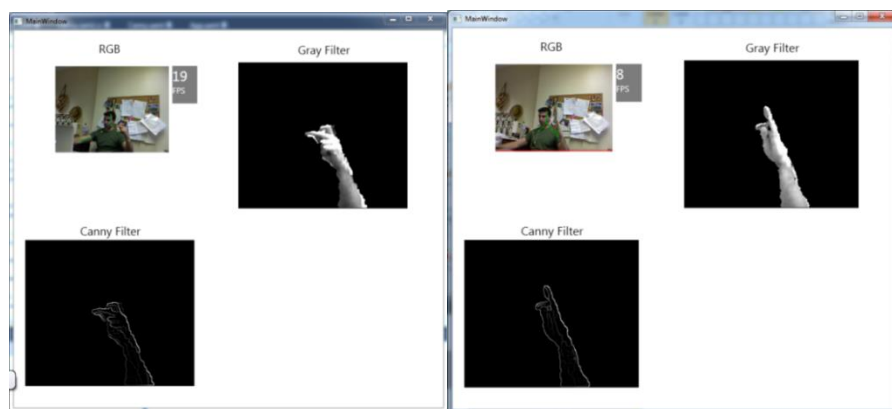


Figura 40 - Detecção de contornos para gesto da letra A (esq.) e gesto da letra “F” (direita)

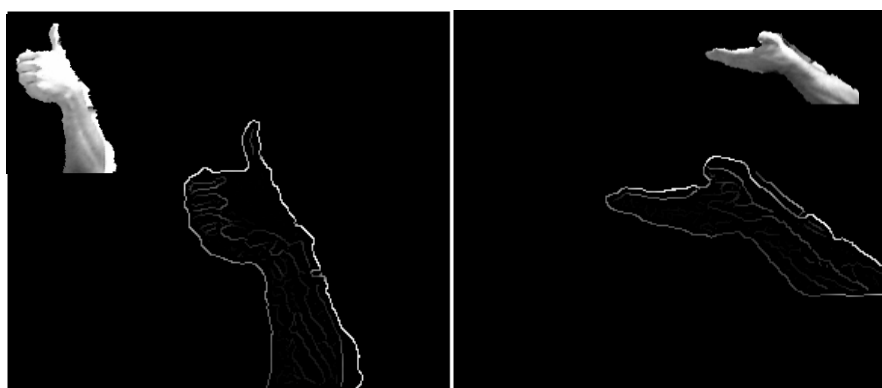


Figura 41 - Detecção de contornos para gesto da letra B (esq.) e gesto da letra “D” (direita)



Figura 42- Detecção de contornos para configuração de mão em garra (usada na LGP, lado esq.) e gesto para letra “V” (dir.)

O algoritmo é eficaz a detetar a silhueta da mão, todavia, relativamente à deteção do contorno dos dedos não se apresenta tão eficaz.

Letra “A” – Há distinção do contorno para 4 dedos, exceto o dedo do polegar que está oculto. É de notar que o algoritmo *Canny* não teve nesta situação a melhor entrada de dados para o gesto, pois o filtro de profundidade cortou alguns dos *pixels* na margem dos dedos mindinho e anelar.

Letra “F” – Os dedos mindinho, anelar e do meio estão sobrepostos, daí ser muito difícil ao algoritmo distinguir os contornos para estes dedos. Além disso, o contorno do resto dos dedos está impreciso e pouco claro.

Letra “B” – A distinção dos contornos não é muito clara ou precisa e volta a acontecer oclusão dos dedos.

Letra “D” – Não há distinção dos contornos em vários dedos.

Configuração mão em garra – Excelente distinção de todos os dedos (configuração de mão usada em LGP).

Letra “V” – Sofre do mesmo problema que no gesto para a letra “A”, em relação ao corte de *pixels*. A distinção está boa apenas para dedo indicador e dedo do meio.

Nas configurações dos gestos existem muitas oclusões, daí o algoritmo desenvolvido por Stegmüller também não foi eficaz na detecção dos dedos, pois as pontas dos dedos estavam quase sempre escondidas. Embora seja possível fazer melhoramentos relativamente à definição de contornos, possivelmente ajustando as configurações de entrada do algoritmo *Canny*, a detecção dos dedos nunca irá ser perfeita. Quando houver oclusão, o dedo não aparece na imagem, e simplesmente não há contorno. Uma das soluções possíveis seria “adivinhar”, ou seja, obter um algoritmo que tenha alguma flexibilidade e consiga prever a posição natural dos dedos quando se encontram ocultos.

Fazer o *tracking* dos dedos com um algoritmo de forma automática teria de ser algo bastante bem feito para ser usado no âmbito deste projeto. O âmbito deste trabalho lida com gestos da LGP que inclui configurações de mãos bastante complexas. Se for desenvolvido um algoritmo simples para tratar esta situação este terá falhas, porque não resolverá as oclusões e consecutivamente não será útil. Por conseguinte, o passo seguinte foi realizar uma abordagem completamente diferente, usando um sistema de *finger tracking* semiautomático, ou seja, o processo de detecção envolve o utilizador para marcar as posições das articulações dos dedos. A seguir descrevo esta abordagem e os problemas encontrados.

3.8 *Finger tracking* semiautomático – aplicação *Pose Estimation*

O *Pose Estimation* tem como objetivo capturar a pose 3D do tronco e mãos. Para as articulações relativas ao tronco a captura é feita de modo automático, recorrendo ao Kinect. Para as articulações relativas às mãos e dedos, a captura é feita de forma assistida pelo utilizador, marcando a posição da mão e todas as articulações de cada dedo numa foto. Após este processo, as coordenadas do gesto são exportadas para ficheiro de modo a serem integradas num avatar no *MotionBuilder*.

Este *software* foi desenvolvido como uma aplicação WPF (*Windows Presentation Foundation*) em C#. O que me levou a ter optado por este tipo de aplicação, foi porque além desta

ser a evolução da classe de aplicações *Windows Forms*, está dotada de mais controlos de utilizador e mais fáceis de usar que o seu tipo de aplicação predecessor. Em seguida apresento os *interfaces* da aplicação:

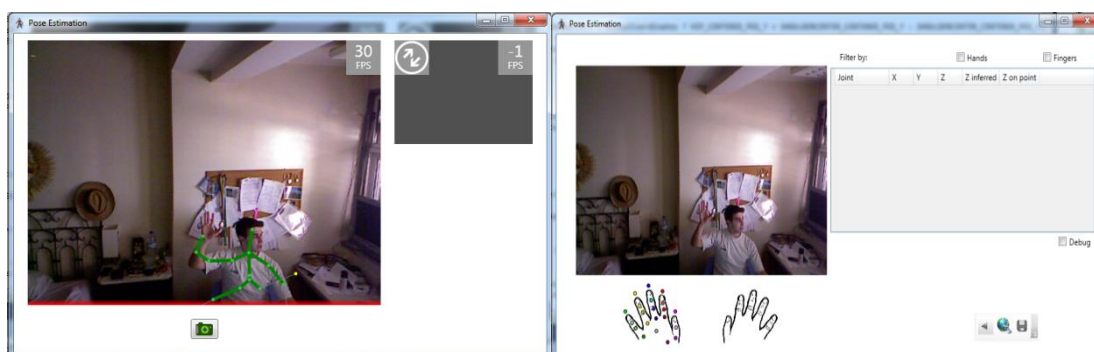


Figura 43 - *Pose Estimation*: ecrã inicial (esq.) e ecrã depois de tirada a foto (direita)

A imagem da esquerda da *Figura 43* representa o ecrã inicial da aplicação onde se vê o utilizador a exibir um gesto para a câmara. Neste *interface*, o utilizador tira a foto fazendo o gesto após o *software* detetar o esqueleto do utilizador. Após tirar a foto no *interface* inicial, o utilizador passa ao *interface* seguinte representado na imagem da direita da *Figura 43*.

Este *interface* permite ao utilizador mapear as articulações dos dedos de cada mão, ou seja, associar os pontos da imagem às articulações dos ossos da falange, falanginha, falangeta e ponta do dedo. Os círculos coloridos em cima da mão situados em baixo da foto capturada pelo sensor, representam os marcadores para cada articulação, dos quais o utilizador pode arrastar cada um para cima da foto. Em alternativa também pode clicar em cima de um ponto qualquer da foto e automaticamente o marcador é posicionado nessa posição segundo uma ordem preestabelecida para cada articulação. Uma vez que o sensor Kinect pode estar longe, como é o caso do exemplo apresentado, e também para resolver a sobreposição dos dedos num gesto, foi adicionada a funcionalidade de *Zoom* à imagem. Na *Figura 44* pode-se ver essa funcionalidade após a marcação das articulações pelo utilizador:

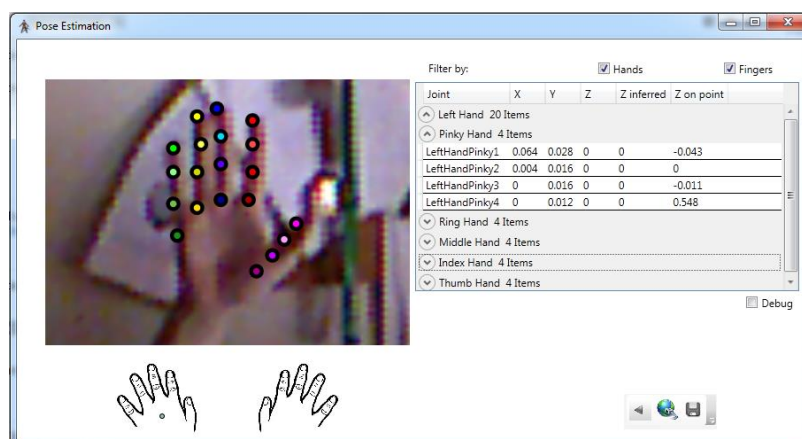


Figura 44 - *Pose Estimation*: interface de marcação de pontos com *Zoom*

O interface *Figura 44* permite ao utilizador visualizar as coordenadas de cada articulação e além disso, considerando que o Kinect não é preciso ao nível da profundidade dos dedos, também permite ao utilizador definir o valor de profundidade de cada articulação através da tabela que se encontra mais à direita. Na *Figura 45*, vê-se o resultado da marcação das articulações dos dedos que o utilizador efetuou na foto, integrado num modelo de esqueleto no *MotionBuilder*.

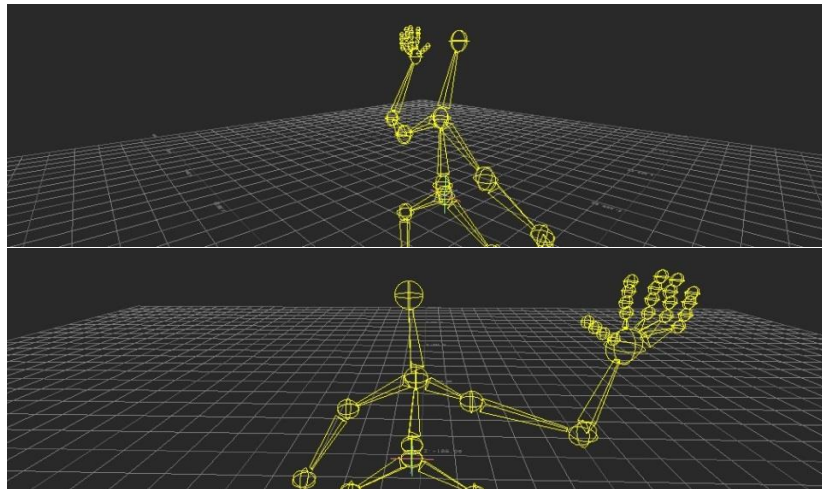


Figura 45 Pose do gesto mão aberta no *MotionBuilder*

As imagens da *Figura 45* representam o mesmo gesto, no entanto, com perspetivas diferentes: a primeira representa uma vista por trás e na diagonal enquanto a segunda representa uma vista de frente. Uma das limitações de marcar numa foto 2D objetos que se encontram num espaço 3D, é a sobreposição desses objetos. Por conseguinte, a funcionalidade de fazer *Zoom* ajudou a lidar com esse problema, como mostra a *Figura 46*:

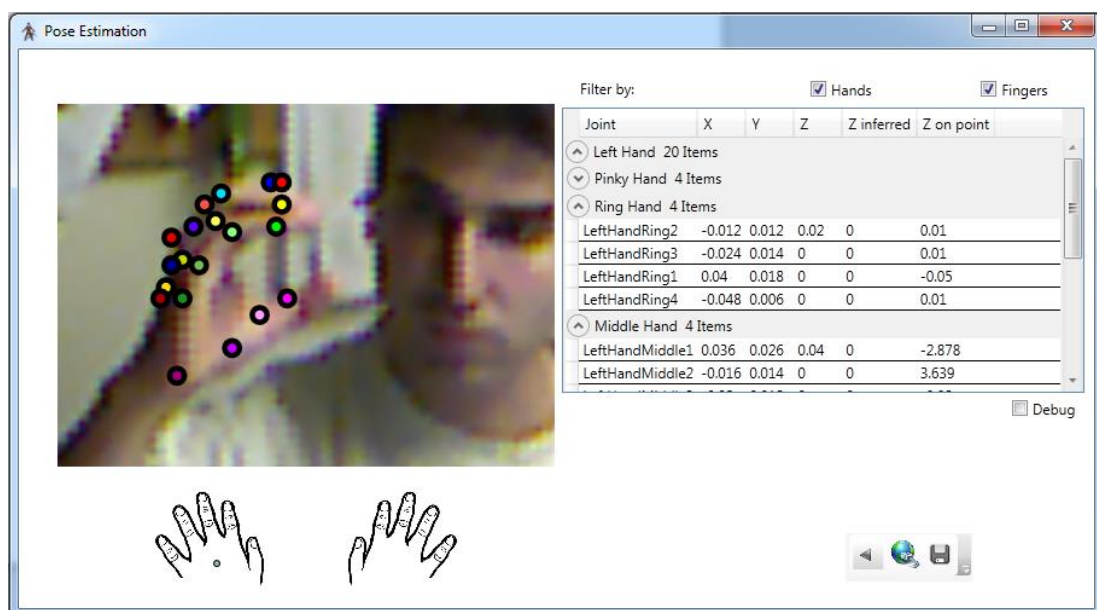


Figura 47 - *Pose Estimation*: marcação de articulações para o gesto relativo à letra “C” em LGP

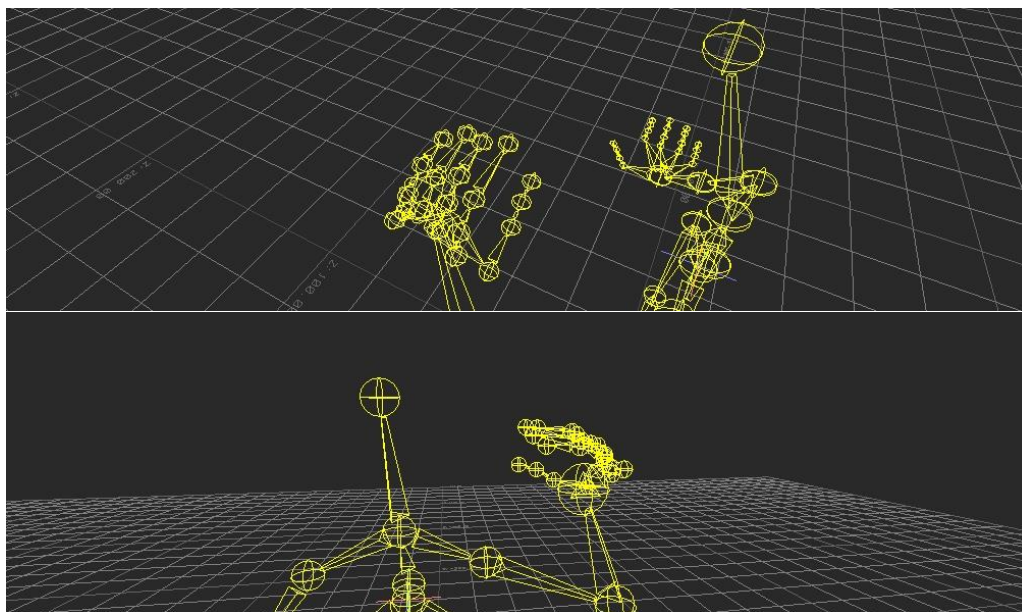


Figura 48 - Pose do gesto para letra “C” em LGP no *MotionBuilder*

A primeira imagem da *Figura 48* representa a vista de cima enquanto a segunda imagem representa a vista de frente. Embora os dedos estivessem totalmente sobrepostos, e a marcação de pontos ter sido difícil dada a proximidade entre as articulações, foi possível integrar o gesto da letra “C” em LGP num esqueleto do *MotionBuilder*.

O próximo passo consistiu em obter os valores de rotação, e perceber como aplicar esses valores num esqueleto. Para em seguida enviar os valores de rotação juntamente com as coordenadas de translação dos dedos para o *MotionBuilder*. Este passo originou o desenvolvimento de um *software* para realizar testes às rotações, descrito na secção seguinte.

Kinect Skeleton 3D

A aplicação permite visualizar um esqueleto a mover-se conforme os movimentos feitos por um humano através do sensor Kinect. As articulações do esqueleto estão representadas através de figuras geométricas como esferas ou cubos. Para gerar gráficos 3D, necessitei de uma *API* gráfica e optei por escolher a biblioteca WPF 3D, que já vem incluída em qualquer aplicação WPF, por causa da sua abstração da complexidade e pelo seu nível de desempenho gráfico suficiente para o que pretendia.

Nesta aplicação o utilizador pode optar por fazer *tracking* em modo normal ou modo sentado. O modo normal faz *tracking* do esqueleto completo e o modo sentado faz *tracking* apenas das articulações do tronco superior.

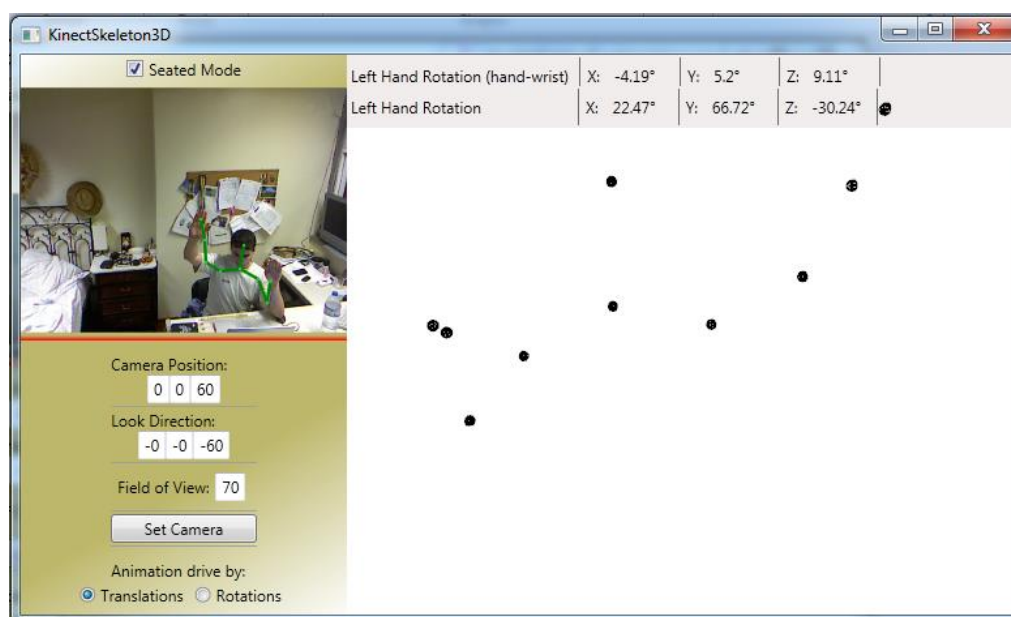


Figura 49 - *Kinect Skeleton 3D* em *seated mode* com esferas

Na Figura 49, o *tracking* está definido para modo sentado, e o esqueleto está representado por esferas. O *Kinect Skeleton 3D* permite representar o esqueleto com cubos também, porque é mais fácil com estes de ver a ligeira diferença na orientação das articulações. Na Figura 50 foi definido o *tracking* de corpo inteiro, através da desmarcação da *checkbox Seated Mode* no canto superior esquerdo.

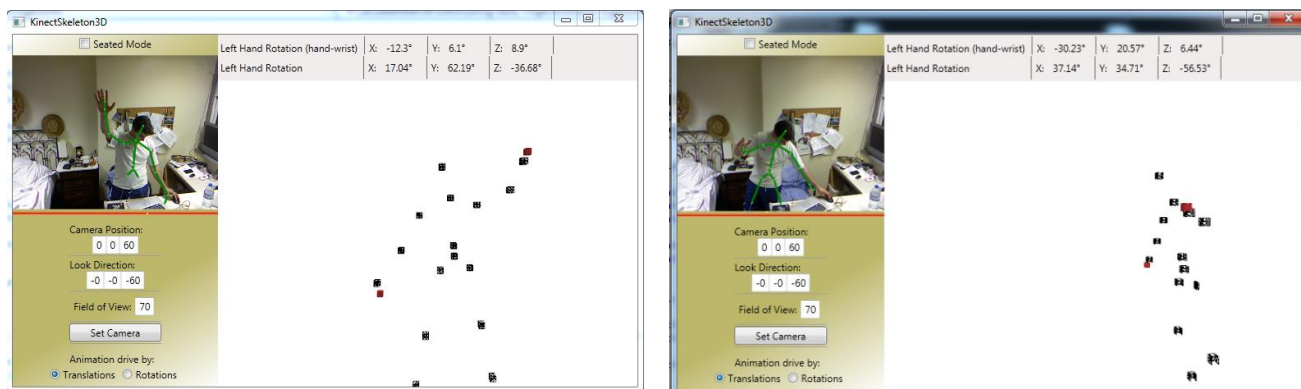


Figura 50 – *Kinect Skeleton 3D*: *full skeleton tracking* com cubos

A Figura 50 mostra outra funcionalidade da aplicação. O esqueleto encontra-se num espaço 3D, logo é possível visualizá-lo em várias perspetivas. Na imagem da esquerda, a câmara está na posição por defeito, ou seja, direcionada para a frente do esqueleto, enquanto na imagem da direita a câmara foi alterada para se posicionar de perfil. Para mover a perspetiva da câmara o utilizador pode usar as teclas direcionais para rodar a perspetiva. As teclas Q, E, A, D alteram a translação da câmara, e as teclas W e S para fazer *Zoom In* ou *Zoom Out*.

TRANSFORM SMOOTH PARAMETERS

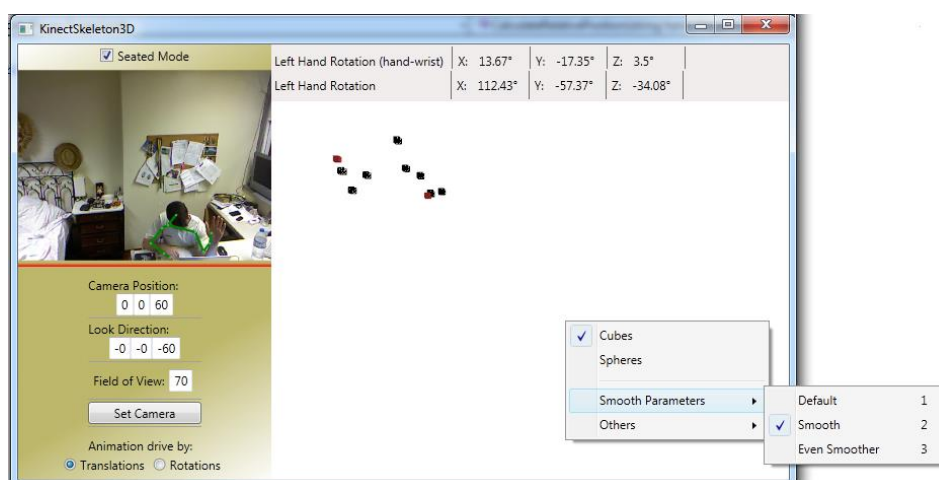


Figura 51 - *Kinect Skeleton 3D*: visualização de opções

Como se pode ver na *Figura 51*, também é possível o utilizador escolher entre 3 tipos de *Transform Smooth Parameters*, os parâmetros responsáveis por definir o nível de fluidez com que as articulações são tratadas pelo sistema de *Skeleton Tracking*. Os 3 tipos são: *Default*, *Smooth* e *Smoother*.

MOVER ESQUELETO

Existem duas modalidades para animar o movimento do esqueleto. O movimento do esqueleto pode ser realizado através dos valores de translação ou rotação das articulações fornecidos pelo Kinect.

ROTAÇÕES

Existem dois tipos de rotações existentes no *SDK* do Kinect: absolutas e hierárquicas. Permite ao utilizador escolher entre elas quando define o tipo de movimento para Rotação. Em seguida descrevo o algoritmo para definir os valores de rotação nas articulações para os dois tipos de rotação:

- *Rotação Absoluta* - Obtém a posição da articulação. Faz o *translate* da figura geométrica e em seguida aplica-lhe a Rotação Absoluta. Faz isto para cada articulação.
- *Rotação Hierárquica* – Obtém a matriz de transformação da articulação pai. Calcula o comprimento do osso, através da subtração entre a posição da articulação pai e a posição da própria articulação. Faz o *translate* da figura geométrica com o valor do comprimento do osso e por fim aplica-lhe a Rotação Hierárquica. Faz isto para cada articulação.

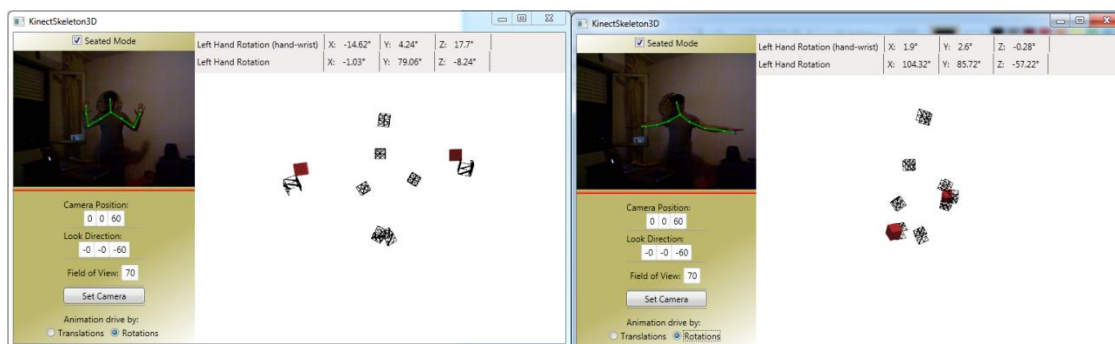


Figura 52 - Kinect Skeleton 3D: esqueleto movido por rotações

Como se pode verificar na *Figura 52*, as rotações tanto no modo de rotação absoluta (imagem da esquerda) como hierárquica (imagem da direita) não corresponderam à pose do gesto feita na imagem de cor RGB.

De modo a chegar à visualização das rotações fornecidas pelo Kinect, era necessário dominar a visualização de translação das articulações porque a definição de rotação para cada articulação usa também o valor da translação. No domínio das translações, o objetivo de mover o esqueleto através delas foi atingido. No entanto, no caso das rotações não.

Análise ao *finger tracking* semiautomático

As limitações do *software Pose Estimation* não são impeditivas ao seu funcionamento, tendo em vista o objetivo a curto prazo do *software* - captura apenas de gestos estáticos (letras). No entanto, a longo prazo, o *software* terá de ser estendido para capturar gestos dinâmicos (palavras) de modo a concretizar um dos principais objetivos da tese – combinação de gestos atômicos em frases de LGP.

Uma das limitações encontradas é a grande quantidade de pontos necessários para definir a posição de um dedo. São precisos 4 pontos para marcar um dedo, logo, para uma mão são precisos 20 pontos. Isto gera um problema quando passarmos à captura de gestos que envolvam duas mãos. Para gestos estáticos que usem duas mãos serão precisos 2 vezes 20 pontos, logo 40 pontos. Todavia, o âmbito desta tese é realizar a combinação de gestos de palavras (gestos dinâmicos) e não apenas letras (gestos estáticos). Portanto, seria necessário marcar em cada foto 40 pontos vezes o número de *frames* que o gesto teria, onde cada *frame* é um instante temporal durante a execução do gesto. É um número muito elevado de marcação de pontos para ser considerado prática e viável esta aplicação. Isto sem falar ainda da segunda limitação que o *software* tem.

Uma vez que o Kinect não tem detalhe suficiente para obter a profundidade com precisão de cada dedo, é necessário que esse valor seja introduzido pelo utilizador. Por conseguinte, além da marcação de pontos por articulação é necessário introduzir um número de valores de profundidade para cada dedo vezes 4 (número de articulações) vezes 2 (número de mãos) vezes o número de *frames* que o gesto tem.

É importante realçar que o *software* foi produto de investigação e a sua linha de desenvolvimento não seguiu uma linha reta. Inicialmente a marcação de pontos para cada dedo apenas precisava de um ponto, visto basear-se na teoria de que o *MotionBuilder* conseguiria definir a posição das restantes articulações do dedo através do seu sistema de *HIK (Human Inverse Kinematics)*.

Contudo, ao longo do desenvolvimento verificou-se que não foi possível definir a posição de um dedo apenas com uma coordenada, logo, isso teve impacto nas limitações do *software* desenvolvido, e por conseguinte, na conclusão da abordagem tomada. O objetivo da segunda aplicação era essencialmente ajudar à concretização da primeira solução. Uma vez que a abordagem teria de ser alterada, e a primeira aplicação não seria continuada, então também não se prosseguiu com o desenvolvimento da segunda.

Conclusão do *finger tracking* semiautomático

O resultado desta investigação deu dois frutos incompletos. Por um lado, um *software* que captura a posição dos dedos e integra esses valores num esqueleto do *MotionBuilder*, através da associação entre pontos da fotografia de uma mão e as articulações de cada dedo. O *software* também captura a pose 3D do tronco, neste caso automaticamente. Por outro lado, um segundo *software* que permite visualizar um esqueleto a mover-se com os movimentos feitos por um humano através do Kinect.

O próximo passo consistiu em exibir a aplicação Solettrar palavras em LGP, para avaliação por utilizadores surdos, e além disso, avaliar frases simples em LGP que surgem pela combinação de gestos atómicos. A animação da configuração da mão far-se-á manualmente por *keyframe* para cada gesto.

3.9 Sumário

Este capítulo consistiu na investigação de várias áreas, nomeadamente, de animação, captura de movimentos, *hand tracking*, processamento de imagem e *finger tracking*.

O capítulo abordou um conjunto de técnicas ou tentativas de usar o Microsoft Kinect[®], para a captura de movimentos aplicada à concretização de gestos de LGP. Numas vezes a animação realizada é totalmente manual, noutras é assistida pelo sensor Kinect. Como este se revelou limitado, adotei um método híbrido, através da captura de movimentos do corpo com o sensor, juntamente com a animação manual da configuração da mão.

Capítulo 4

Geração e animação do Avatar

Neste capítulo apresenta-se a componente do trabalho realizado, relativa à geração e animação das personagens virtuais que se integraram na aplicação final. Além disso, complementa a investigação anterior sobre a captura de movimentos, mas desta vez, capturando a componente em falta à interpretação da LGP, a expressão facial.

Em seguida, cada fase do trabalho descrita em seguida, é comparada com a iteração do processo de desenvolvimento respetiva.

4.1 Protótipo base - primeira iteração

A primeira iteração consistiu no desenvolvimento e avaliação por utilizadores de um protótipo inicial. Este protótipo tem duas funcionalidades principais: permite *Soletrar palavras em LGP* – criação de *plugin*; e permite visualizar um conjunto de frases em LGP feitas a partir da combinação de gestos atómicos.

A primeira funcionalidade do protótipo já tinha sido desenvolvida anteriormente (3.3) e permite ao utilizador escrever uma palavra, e em seguida visualizar um avatar a executar o gesto em LGP para cada letra do que foi escrito. Fizeram-se algumas modificações à ferramenta desenvolvida anteriormente para contemplar cinco palavras que o utilizador não sabe à partida quais são, mas que terá de adivinhar o seu significado. As palavras escolhidas cobrem a maior parte das letras incluídas no abecedário: *embora, Portugal, favor, caminhar e ajudas*.

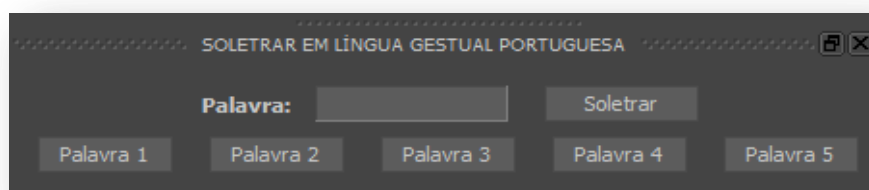


Figura 53 - Soletrar palavras em LGP com palavras predefinidas (esq.) e Avatar utilizado (dir.)

4.1.1 Visualizar frases em LGP

A segunda funcionalidade do protótipo consiste em apresentar ao utilizador 6 frases em Língua Gestual Portuguesa que foram geradas pela combinação de gestos atómicos de palavras. Ao contrário da primeira funcionalidade do protótipo (Soletrar palavras em LGP) do qual o método de animação foi realizado manualmente por *keyframe*, agora recorri à captura de movimentos para animar o corpo e à animação manual para a configuração da mão. Esta funcionalidade pretende ir mais longe que a primeira e apresenta frases em vez de palavras.

O conjunto de expressões é composto por uma frase de saudação, quatro afirmativas e uma negativa:

Frase em LP	Frase em LGP
Olá. Bom dia. Tudo bem?	<i>“OLÁ BOM DIA TUDO BEM?”</i>
Eu vou à escola.	<i>“EU ESCOLA IR”</i>
Ele tem um gato.	<i>“ELE GATO TER”</i>
Ele foi a Évora.	<i>ELE ÉVORA IR</i>
Eu não vou à escola.	<i>EU ESCOLA IR NÃO</i>
Eu tenho uma gata.	<i>EU MULHER GATO TER</i>

Tabela 3 – Frases exemplo em LP e respetiva glosa

A escolha destas frases teve em consideração certas palavras-chave como: “*passado*”, “*não*” e “*mulher*”. Estas palavras permitem compor frases cujo tempo verbal está no passado, frases negativas ou marcar o feminino de uma palavra. Note-se que o uso da palavra “*mulher*” antes de “*gato*” forma a palavra “*gata*”, e a palavra “*passado*” ou “*futuro*” depois de um verbo, marca o correspondente tempo verbal do verbo. A palavra “*não*” no fim da frase nega uma frase declarativa em LGP.

Em seguida, foram capturadas as palavras ou os gestos atómicos que compõem as frases definidas. Para ser possível a combinação de gestos, foi necessário retirar da animação o segmento inicial e final de cada gesto., ou seja, definiu-se como início e fim de cada animação, o instante em que as mãos do avatar passam o nível da cintura. Este processo foi feito sem prejudicar a interpretação do significado do gesto em LGP.

Para a captura de gestos usou-se o sensor Kinect e o *software Brekel Kinect* que permitem transpor os movimentos humanos para um avatar no *MotionBuilder*. O processo de captura dos

gestos foi o mesmo utilizado anteriormente e encontra-se descrito em *Animação de gestos de palavras de LGP*. O passo seguinte foi a composição das frases.

Composição de Frases

Nesta fase foi necessário juntar os gestos atômicos ou palavras numa só animação. Deste processo resultaram 6 animações, logo, 6 frases. Usando o menu *Story* do *MotionBuilder* foi possível criar várias *Tracks* (contentores de animações), e usar cada uma para formar uma frase. Em cada *Track* juntou-se todos os gestos atômicos ordenados segundo as regras gramaticais da LGP.

Ao concatenar as animações é preciso sobrepor a animação de um gesto com a animação do gesto anterior. Também chamado de *blend* entre gestos, este processo consiste na interpolação entre as duas animações ou gestos. O *MotionBuilder* trata do processo de *blend* automaticamente gerando uma animação fluída e consistente. O nível de consistência e fluidez está dependente de dois factores: da diferença entre as duas animações, logo, neste caso particular, depende da diferença da posição das mãos entre as animações; e do tempo definido de *blending*, isto é, do tempo definido para a sobreposição entre as duas animações. Estes dois fatores foram ajustados para cada animação, de modo a obter cada animação com a melhor qualidade possível.

Foram capturadas também duas animações acessórias para marcar o início e o fim do gesto. Uma vez que cada gesto começa e acaba com as mãos ao nível da cintura, juntou-se a cada frase, uma animação inicial e uma animação final. A animação inicial faz a passagem do estado inicial do avatar (posição de descanso) para o início do gesto enquanto a animação final faz a passagem do fim do gesto para a posição de descanso do avatar.

A estrutura utilizada para compor as frases foi sujeito-objeto-verbo (SOV) segundo a mesma regra utilizada no dicionário de Língua Gestual Portuguesa (Baltazar 2010).

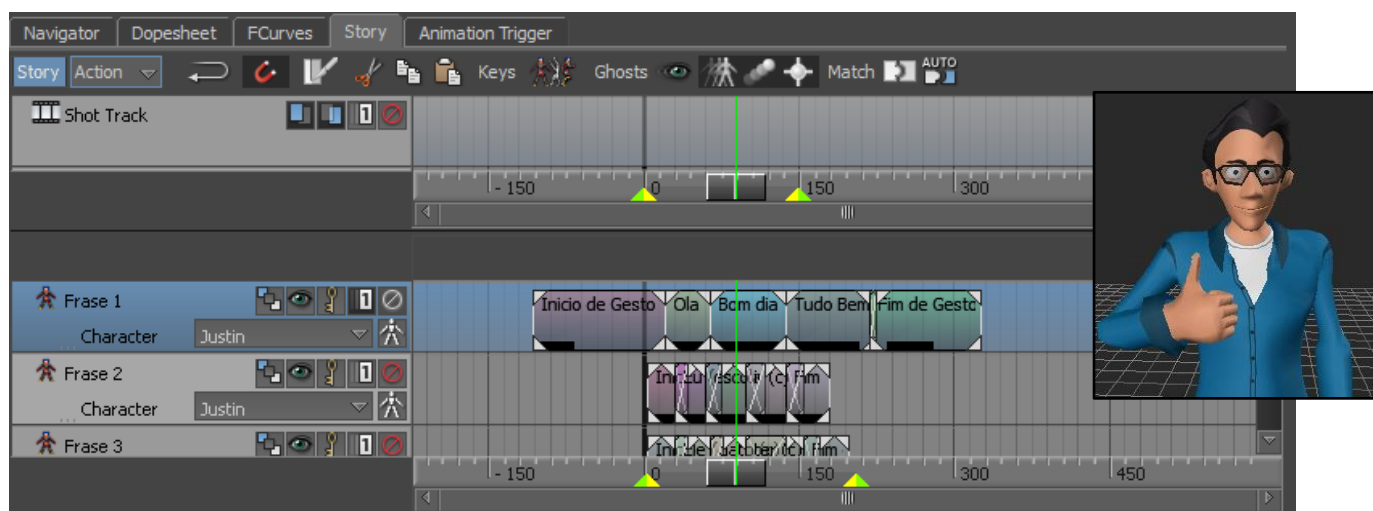


Figura 54 – Menu *Story* do *MotionBuilder* (esq.) e avatar utilizado (direita)

Na imagem do lado esquerdo da *Figura 54*, pode ver-se o menu *Story* que permite compor frases. Cada *Track* tem o nome de *Frase 1*, *Frase 2* e assim sucessivamente. Em cada *Track* residem os gestos ordenados sequencialmente segundo a sintaxe da LGP. Na *Frase 1* podemos ver os gestos “*Início do Gesto*” seguido de “*Olá*”, “*Bom dia*”, “*Tudo bem*” e “*Fim do Gesto*”. Para efeitos demonstrativos não se realizou o *blending* dos gestos para se interpretar claramente o que são os retângulos coloridos (as palavras), porém na *Frase 2* já se pode visualizar o *blending* entre os gestos e constata-se que graficamente os retângulos coloridos sobrepõem-se ou interseccionam-se entre si. Depois de desenvolvidas as duas funcionalidades descritas, passou-se à sua avaliação com utilizadores surdos.

4.1.2 Avaliação com surdos na Associação Portuguesa de Surdos

A Associação Portuguesa de Surdos (APS) é uma organização que suporta e apoia a comunidade de surdos em Portugal. Tem como fim defender os direitos legais individuais e comunitários dos surdos. Desloquei-me à sede desta associação para avaliar o 1º protótipo com utilizadores surdos.

A avaliação foi efetuada por duas surdas entre os 25 e 45 anos. Uma das surdas é professora e ensina Língua Gestual a crianças de 1 a 3 anos. A associação também dispõe de uma intérprete para traduzir a conversa entre nós. No decorrer da conversa inicial fiquei a saber que eles não conhecem muitas aplicações de aprendizagem para Língua Gestual Portuguesa e não mexem muito com computadores. As aplicações mais utilizadas por surdos são DVDs como dicionários de Língua Gestual, DVDs que contam histórias para crianças e DVDs para pais de filhos surdos para aprenderem a comunicarem com os seus filhos.

Objetivos

A avaliação do primeiro protótipo teve três objetivos principais:

1. Testar a qualidade gráfica das animações e dos avatares;
2. Testar a facilidade de interpretação das animações;
3. Fomentar a discussão para a criação de uma aplicação útil de LGP

O teste com utilizadores foi dividido em duas partes.

A **primeira parte** envolveu a utilização da aplicação *Soletrar Palavras* em LGP e pretendeu testar o desempenho do nosso intérprete virtual a soletrar palavras em LGP.

Na **segunda parte**, apresentou-se um conjunto de frases e pretendeu-se avaliar o desempenho de um avatar a traduzir frases predefinidas para LGP.

Métricas

As métricas para a avaliação basearam-se em dois questionários, um para cada parte da avaliação. Ambos os questionários medem o desempenho de uma funcionalidade particular do protótipo. O primeiro questionário recolheu as repostas relativas ao teste da componente de Soletrar palavras em LGP enquanto o segundo recolheu o *feedback* relativo à visualização de frases em LGP. Cada questionário contém perguntas de repostas abertas e perguntas com escala *Likert* de 3 níveis para avaliar a facilidade com que o utilizador interpretou os gestos.

Questionários

Os questionários usados na avaliação podem ser consultados no *Anexo 3* para o primeiro teste e no *Anexo 4* para o segundo teste.

Resultados

Começámos o teste e os surdos ficaram entusiasmados pelo avatar da *Figura 53*. Ao apresentar a primeira palavra (“*embora*”), os surdos sentiram muita confusão e não conseguiram perceber. Queixaram-se que a animação estava muito rápida e além disso, não conseguiam visualizar bem o contraste entre o braço e a mão do avatar. Diminuir a velocidade de animação, tornou possível a interpretação. Após 4 tentativas conseguiram interpretar a primeira palavra.

Passámos à segunda palavra que era “*Portugal*”, que foi reconhecida ao fim de duas tentativas. Ao longo do teste identificaram outro problema. Nas letras cujo gesto implica direcionar a mão para baixo, como por exemplo, nas letras “*M*” e “*N*”, a animação não estava correta, pois o braço afastava-se demasiado da posição correta. A posição correta é posicionar a mão no centro do corpo à frente dos pulmões.

No segundo teste, o avatar utilizado era diferente (*Figura 54*). Os surdos confirmaram que este avatar era mais adequado pois era mais fácil interpretar os gestos devido às mãos estarem mais expostas e contrastantes do resto do corpo. De seis frases apenas três foram reconhecidas. Das frases reconhecidas a primeira foi reconhecida mais facilmente que as restantes, enquanto as outras duas tiveram de ser repetidas.

Foram identificados 6 causas para o não reconhecimento:

- Construção frásica errada;
- Regionalismos: execução de gestos que são utilizados numa região diferente da região donde os surdos residem;
- Má qualidade na animação para alguns gestos;
- Falta de expressão facial do avatar;
- Alternância da mão dominante durante a execução dos gestos;

Cinco das seis frases utilizadas usavam a estrutura frásica SOV, no entanto, os surdos afirmaram que essa não era a ordem correta para os elementos da frase, mas sim OSV (objeto-sujeito-verbo). Outro motivo foi a utilização de gestos como “gato” e “ir” que foram executados como se faz no Porto e que são diferentes dos que os surdos usam habitualmente na região de Lisboa. Outra causa foi que alguns gestos como “ir” não estavam bem animados, e no caso do gesto “ir” o gesto entrava em 3 frases. Para os gestos “Escola” e “Évora”, a mão dominante era a esquerda, enquanto para o resto dos outros gestos a mão dominante era a direita. Não é correto fazer a alternância da mão dominante durante a execução dos gestos.

Conclusão dos resultados

Foi muito importante o contacto com a associação. O teste com utilizadores reais evidenciou os pontos fracos e fortes, e fez-me ver que é necessário um acompanhamento próximo da parte dos surdos ao longo do trabalho.

O avatar usado no primeiro teste é desadequado e foi descartado, pois quanto mais visíveis as mãos estiverem, melhor será para o surdo decodificar com sucesso a mensagem que o avatar está a tentar transmitir visualmente. Concluí que seria necessário refazer algumas letras, especialmente as letras que implicam que a mão esteja virada para baixo e também reposicionar a posição do braço nessas letras.

Em relação ao segundo teste, verificou-se que algumas expressões como “ir” têm de ser refeitas para atingir uma animação com mais qualidade. Além disso verificou-se que a ordem dos elementos na frase não estava correta. Não existe consenso na LGP quanto qual a ordem predominante (Wikipedia 2007): alguns linguistas afirmam ser SOV, outros afirmam ser OSV. Portanto, vou adotar daqui adiante a construção frásica OSV, a referida pelos surdos na avaliação.

Durante a reunião, foi discutido com uma surda cuja área de trabalho é o ensino de LGP a crianças surdas, algumas necessidades da sua área. Ela referiu que na sala de aula a utilização de um avatar para ensinar gestos de LGP cativaria o interesse das crianças pela aprendizagem. Durante a conversa, discutiram-se duas ideias para aplicação de um avatar:

1. Avatar que contasse histórias a crianças;
2. Avatar que traduzisse palavras para LGP como cores, profissões, etc.

O ponto 2 foi a ideia que originou a aplicação final desta tese. Antes de passar à descrição da aplicação era necessário integrar as alterações suscitadas e gerar um avatar interessante para o contexto infantil que tivesse em conta as opiniões referidas durante a reunião.

4.2 Novo avatar - segunda iteração

A segunda iteração vem no seguimento da avaliação do protótipo inicial. Face às alterações suscitadas pelos surdos durante a avaliação, a segunda iteração consistiu em obter um avatar com

mais qualidade e mais adequado à execução de gestos no âmbito de uma aplicação infantil. Posteriormente, procedeu-se à reavaliação do resultado por utilizadores surdos.

4.2.1 Conceção de novo Avatar

Ao contrário dos avatares utilizados até aqui, este avatar foi criado por mim. Para facilitar a criação foram utilizadas várias ferramentas para gerar e trabalhar o avatar. As ferramentas utilizadas foram: *MakeHuman*; *DAZ Studio* e *Blender*. *Blender* (Blender Foundation 2011) é um *software open-source* para modelação e animação 3D. *MakeHuman* (Makehuman TM 2011) é uma aplicação desenhada para a geração de humanos virtuais fotorealísticos. *DAZ Studio* (DAZ 3D Inc 2011) é uma ferramenta de ilustração e animação 3D sem recorrer à complexidade da modelação.

O *MakeHuman* permitiu gerar facilmente duas personagens virtuais. Tendo em conta que a aplicação final será focada para crianças, então decidi criar avatares de crianças. Com a finalidade de haver diversidade racial, foram representados duas personagens: uma menina chinesa e um menino africano. Depois de criados, estes foram exportados e importados para o *Blender*. A exportação direta do *MakeHuman* para o *MotionBuilder* apresentou alguns erros ou incompatibilidades pelo que tive de recorrer à ferramenta *Blender* (*Figura 55*) como aplicação intermédia que permitiu corrigir os problemas encontrados.

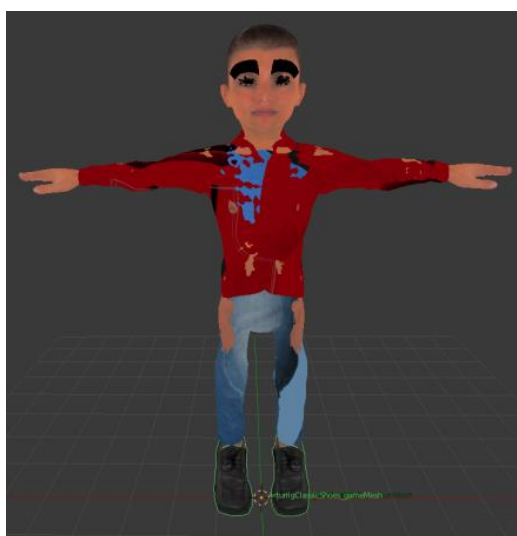


Figura 55 – Avatar com problemas na importação

As incompatibilidades encontradas como se podem ver na *Figura 55* foram: as texturas não foram bem carregadas, designadamente, as pestanas e as sobrancelhas apareceram a preto; e a pele do avatar trespassava a roupa do próprio. Em relação ao primeiro problema, a solução passou por apagar todos os vértices que representavam as pestanas e as sobrancelhas, conseguindo assim eliminar as figuras geométricas a preto. Para resolver a segunda situação, tentou-se algumas abordagens, uma das quais foi fazer o *Vertex Modeling* que consiste na alteração da posição dos

vértices do modelo. No entanto, este método demonstrou-se pouco prático, pois era muito demorado e pouco eficaz, visto que era necessário alterar uma grande quantidade de vértices e por sua vez, algumas alterações à malha provocavam irregularidades indesejadas na forma/dimensão do personagem. O método utilizado passou por usar um *Mask Modifier*. Este método consiste em esconder um conjunto de vértices. Criou-se um *Vertex Group*, que é basicamente como o nome indica um grupo de vértices, agrupou-se apenas os vértices responsáveis pelos pontos da malha que estão por baixo da roupa e aplicou-se um modificador do tipo *Mask* a esse grupo. Depois de corrigido os problemas referidos, decidi criar um terceiro avatar. Neste caso foi criado uma menina caucasiana recorrendo ao *software DAZ Studio*. A razão para ter usado outra ferramenta, foi a necessidade de criar um avatar com uma aparência completamente diferente dos personagens anteriores, tanto ao nível de estrutura física, como roupa, e por isso não se recorreu ao *MakeHuman* como nos outros avatares. A exportação do *DAZ Studio* para *MotionBuilder* não trouxe nenhum problema.

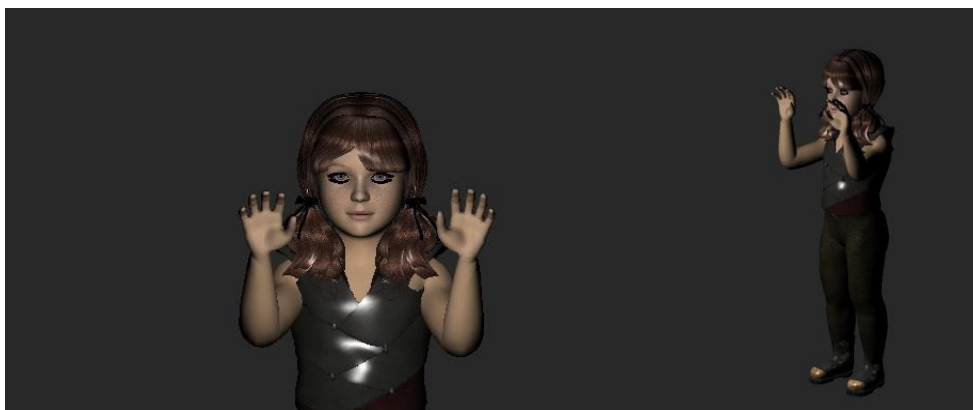
Resultados



(a)



(b)



(c)

Figura 56 – Foto dos avatares: (a) *Artur* - (b) *Mimi* - (c) *Nucha*

Foram concebidos três avatares já com esqueleto de diferentes raças, e a sua importação foi testada com sucesso pelo *software* de animação *MotionBuilder*. Os três avatares estavam preparados para serem animados. A próxima fase foi decidir qual dos três seria o melhor avatar na perspetiva do público-alvo. Então contactei de novo a Associação Portuguesa de Surdos para saber qual dos três seria o melhor modelo no papel de comunicador de LGP no âmbito de uma aplicação infantil. A APS voltou a contactar os surdos que estiveram presentes na avaliação do primeiro protótipo e validaram que o avatar *Nucha* seria o mais indicado para o objetivo.

O próximo passo consistiu na última iteração do projeto e teve três objetivos: animar o avatar escolhido com várias palavras e frases; investigar a captura e animação facial; desenvolver uma aplicação que expusesse as funcionalidades do avatar como intérprete gestual.

4.3 Novos gestos, captura facial e aplicação - terceira iteração

A terceira iteração envolveu a produção de novos gestos para estender o vocabulário existente, mas desta vez, tendo em conta o *feedback* recolhido na primeira avaliação com surdos. Uma das recomendações era a falta de expressividade, e esse campo foi abordado através de alguns testes à captura facial e na sua integração numa animação de corpo e mãos previamente elaborada. Por fim, esta iteração finalizou com a elaboração clara do conceito de uma aplicação e a sua implementação de modo a exhibir as funcionalidades de animação conseguidas até aqui.

4.3.1 Produção de novos gestos

A maioria dos gestos existentes possui uma qualidade razoável. No entanto, dois usavam como mão dominante a esquerda, distinta da mão utilizada na maioria dos gestos. Com este cenário, emergiam duas opções: voltar a produzir os dois gestos utilizando a direita como mão dominante, ou passar a maioria dos gestos para a mão esquerda. Como capturar e produzir a animação dos dedos é um processo demorado, a primeira opção seria muito mais fácil e prática. A desvantagem é que os gestos produzidos foram frutos de um nível de experiência baixo de

animação, e não refletiam a qualidade por mim desejada. Decidi capturar de novo todos os gestos, exceto os dois que já se encontravam com a mão dominante a esquerda, e produzir para cada um a animação de mãos e dedos. Além disso, capturei mais 22 gestos novos.

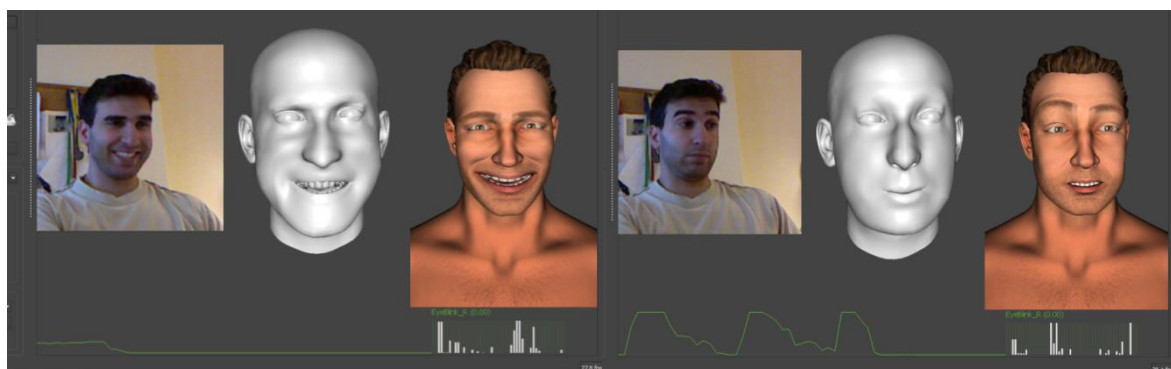
Os novos gestos incluíram gestos para novas cidades, animais, cores, profissões, frutas, verbos, pronomes interrogativos, entre outros. A escolha das novas palavras foi pensada já com a ideia de quais as frases que poderiam ser feitas com a combinação dessas palavras. Além disso, para estender os conceitos de sintaxe do primeiro protótipo (marcação de gênero, negação de frase, tempo verbal), foram capturados pronomes interrogativos para gerar frases interrogativas e capturei a palavra “*muitos*”, para gerar palavras no plural em LGP. Relembro que uma das formas de marcar o plural em LGP, é pela adição da palavra “*muitos*” depois do substantivo.

A produção de 20 gestos desde a captura até ao gesto final, demora cerca de 10 horas, sendo despendida uma média de meia hora para cada gesto. O passo seguinte envolveu fazer um teste à captura facial e integrá-la num avatar.

4.3.2 Captura facial

Para a capturar as expressões faciais foram testados duas ferramentas: *Brekel Kinect Pro Face* (BKPF) e *Faceshift*. Ambas exportavam os dados para o formato FBX de modo que facilitava a importação direta para o *software* de animação *MotionBuilder*. A qualidade de *tracking* do BKPF não é tão boa nem estável como a do *Faceshift*, pelo que este motivo a ferramenta escolhida para a captura foi o *Faceshift*.

O *software* utiliza um algoritmo de aprendizagem (*machine learning algorithm*) para otimizar a qualidade do *tracking* a um utilizador específico, pelo que precisa inicialmente de ser treinado através da execução de uma série de expressões faciais. Após treinar o *software* para se adaptar à minha face, procedi à captura de expressões como feliz, surpreendido e zangado demonstradas na **Figura x**.



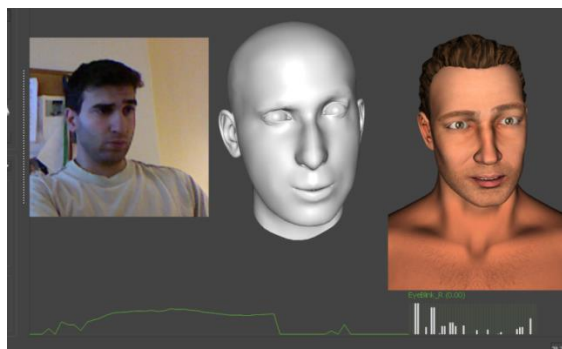


Figura 57 – *Faceshift*: expressões faciais: feliz (esq.), surpreendido (direita), zangado (baixo)

Na *Figura 57* nota-se que a pose facial para as expressões feliz e surpreendido são bem capturadas, mas expressão de zangado não é bem capturada. Algumas expressões não são tao bem capturadas como outras. A *Figura 57* demonstra que o *Faceshift* consegue apresentar um avatar e até importar um avatar externo e animá-lo com os dados da captura em tempo real. No entanto, os avatares existentes por defeito no *software* apenas têm face e peito, pelo que importou-se um avatar externo para testar a animação facial no *MotionBuilder*.

O *Faceshift* permite utilizar *blendshapes* para animar as expressões faciais. Os avatares utilizados até ao momento não tinham *blendshapes* e uma vez que a plataforma online *Project Pinocchio* (Autodesk 2013) gera muito facilmente avatares já com estes elementos, então decidi criar um avatar com esta ferramenta e importar para o *Faceshift*. Para integrar um avatar externo no *Faceshift* é necessário mapear as *blendshapes* do avatar com as *blendshapes* que o *software* está à espera. O *software* tem um editor (*Figura 58*) que permite fazer esta associação, onde se vê de um lado a *blendshape* do software e a sua representação gráfica na face de um modelo humano, e do outro lado, as *blendshapes* do avatar e a sua representação gráfica na face do avatar.

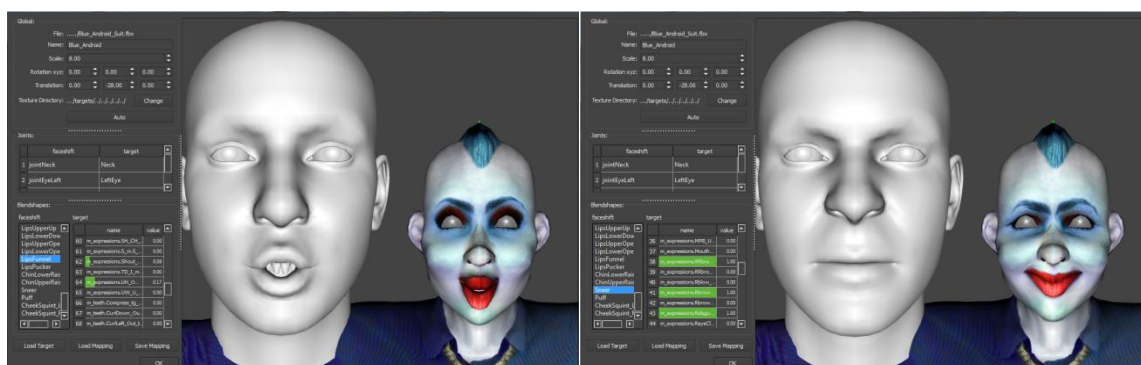


Figura 58 – *Faceshift*: mapeamento de *blendshapes* entre avatar e *software*

Este processo é demorado e complicado porque existem 48 *blendshapes* e 90% das *blendshapes* do *software* não correspondem às do avatar. A solução passa por jogar com os valores das *blendshapes* no avatar até que a pose facial do modelo humano do *Faceshift* corresponda com a pose facial do avatar.

Depois de realizado o processo anterior, capturou-se uma expressão facial de um sorriso e exportou-se para o *software* de animação. Ao ser importado no *MotionBuilder*, o avatar apresentou os olhos a preto (Figura 58). Julgo que o *software* não lida bem com materiais com transparências, então a solução passou por apagar os dois elementos que tinham transparência. O próximo desafio foi integrar a animação facial com uma animação do corpo já existente.

Para isso importei o gesto “*Olá*”, previamente utilizado no protótipo inicial. Associei o gesto ao novo avatar, e através do menu *Story* consegui juntar a animação facial à animação do corpo, sincronizando o instante inicial e final de cada uma. Na *Figura 59* vemos o gesto “*Olá*” em dois instantes diferentes com animação facial incorporada.



Figura 59 – Gesto “*Olá*” em LGP: início do gesto sem expressão facial (esq.) e a meio do gesto vê-se a expressão facial

4.3.3 Aplicação

A aplicação concebida chama-se “*Os meus primeiros gestos*” e está descrita no próximo capítulo.

4.4 Sumário

Este capítulo enumerou as três iterações que descrevem a metodologia utilizada para realizar os objetivos desta tese. O capítulo começou por abordar o desenvolvimento de um protótipo inicial e a sua avaliação por utilizadores. Em seguida, o capítulo descreveu o processo de criação de três avatares, onde um deles foi escolhido por surdos para colmatar as falhas do avatar utilizado anteriormente. Por fim, apresentou-se o teste feito à captura facial, mostrando o processo desde a captura da expressão facial, passando pela sua integração num avatar, até à expressão facial ser incorporada numa animação efetuada anteriormente para o corpo, incluindo mãos e dedos.

Capítulo 5

Os meus primeiros gestos

O trabalho realizado previamente pode ser aplicado em vários contextos possíveis. Este capítulo inicia-se pela motivação que levou à elaboração de um conceito particular de aplicação.

O capítulo descreve todo o processo de desenvolvimento de uma prova de conceito, uma aplicação *Web*, que pretende introduzir conceitos básicos de LGP e que foi desenhada principalmente para crianças.

5.1 Motivação

Antes de passar ao desenvolvimento da aplicação, fiz uma breve pesquisa sobre como as crianças surdas são ensinadas na escola. Li acerca da dificuldade que os alunos tinham em aprender o português escrito. Li também que os métodos de ensino da componente escrita tinham em comum a associação de imagens aos conceitos ou ideias que se queriam transmitir ao surdo (Camargo 2011) (Baptista 2010). Concluí que uma aplicação para ensinar conceitos básicos de Língua Gestual Portuguesa, tem necessariamente de correlacionar o conceito que se quer ensinar - a sua palavra escrita – com a sua representação visual e o gesto respetivo em LGP.

Durante a reunião com surdos na Associação Portuguesa de Surdos (*Avaliação com surdos na Associação Portuguesa de Surdos*), surgiu a ideia de criar uma aplicação que introduzisse conceitos básicos de LGP a crianças, com vários temas como cores, profissões, etc. Como diz Deusdado, a consulta de dicionários para o ensino da LGP não é suficiente, sendo que o aluno deve um papel mais ativo e no processo de aprendizagem e não meramente só visualização de gestos. Com esse objetivo, decidi incluir uma componente, distinta dos dicionários tradicionais, que teste os conhecimentos do utilizador, através de um jogo simples de associação entre palavras e gestos, e uma forma divertida de compor palavras em frases de LGP.

5.2 Os meus primeiros gestos

Esta aplicação pretende introduzir conceitos básicos de LGP. Consiste num dicionário temático que permite a consulta de palavras e a sua tradução em gestos de LGP. Tem temas como

cores, profissões, animais, frutas, cidades, verbos e pronomes pessoais e interrogativos. O tradutor é um avatar que comunica em Língua Gestual Portuguesa os conceitos a serem traduzidos. “Os meus primeiros gestos” também contempla a apresentação de vídeo com humanos, no caso de não existir uma animação com um avatar para traduzir um gesto. Tem gestos para palavras como: “eu”, “não”, “ir”, “comprar”, “passado”, “bombeiro”, “laranja”, “Lisboa”, “gato”, “cão”, “vermelho” e “azul” para nomear apenas algumas. Na Figura 60 pode-se ver o avatar a traduzir para gestos de LGP a palavra “verde”.

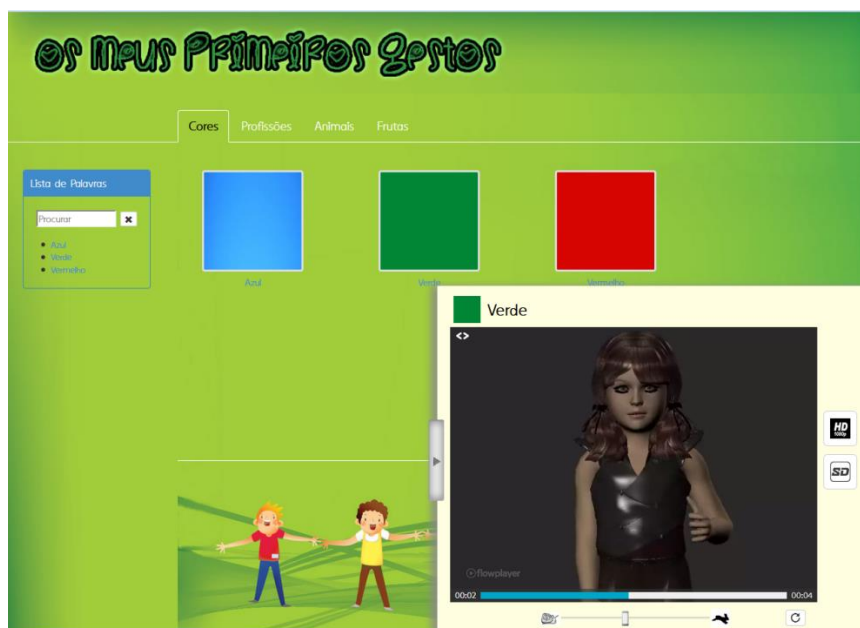


Figura 60 – Ver um gesto na *interface* principal

Também se pode compor frases com palavras do dicionário. É possível arrastar uma palavra de cada vez e compor uma frase com a sintaxe LGP. Por exemplo, um utilizador pode compor a frase em LGP “GATO VERDE”, ou seja, “O gato é verde” em LP. O avatar então mostra a sequência de gestos para essa frase mais a frase em LP correspondente. A composição de frases é de estrutura simples, mas está focada para ensinar alguns conceitos gramaticais como: a marcação de género, a marcação de tempo verbal, o singular/plural, a construção de frases negativas e interrogativas. No compositor (Figura 61), ao combinar as palavras “mulher”, “gato” e “verde”, obtém-se a frase “A gata é verde” em LP e a visualização dos gestos correspondentes. “Os primeiros gestos” também inclui um modo jogo ou exercício (Figura 62), para de uma forma mais didática testar os conhecimentos do utilizador. O utilizador escolhe uma palavra e são apresentadas várias animações de gestos, sendo que apenas uma corresponde ao gesto dessa palavra. O utilizador terá de adivinhar qual é a animação correta.

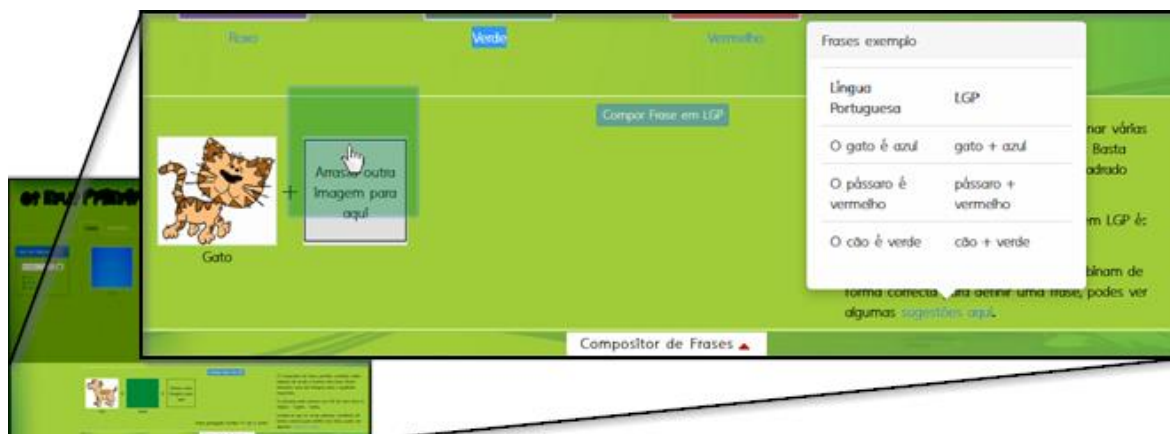


Figura 61 - Compositor de frases (em baixo)

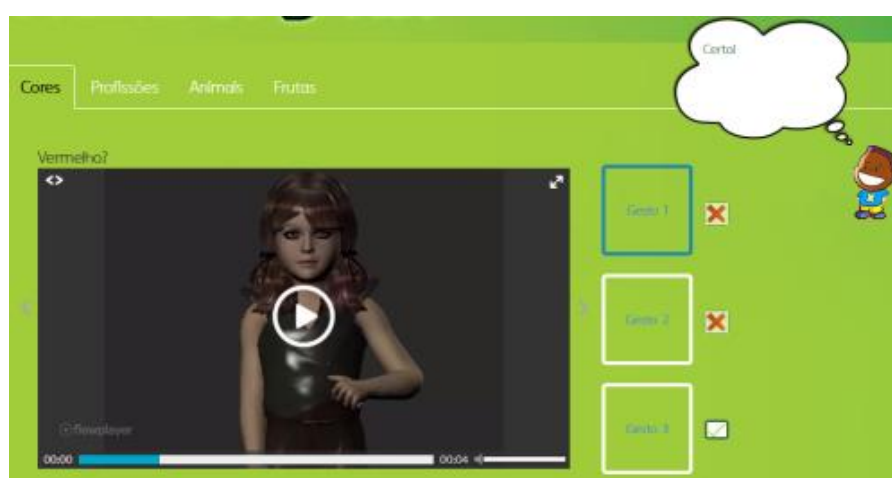


Figura 62 - Adivinhar gesto

Esta prova de conceito foi desenhada para ser extensível, ou seja, permite facilmente a gestão dos seus conteúdos através de uma *interface*, ilustrado na *Figura 63*, para adicionar, remover e editar palavras, gestos, frases e utilizadores por um administrador.



Figura 63 – Gestor de Conteúdos

5.3 Público-alvo

Esta aplicação destina-se a dois públicos-alvo distintos. Por um lado, destina-se a surdos, principalmente crianças, que tenham LGP como primeira língua, mas que sintam dificuldade em associar o significado das palavras à sua representação escrita. Por outro lado, destina-se a ouvintes do qual se pressupõe que a sua primeira língua seja o português, mas que não tenham conhecimentos de língua gestual portuguesa e com o uso da aplicação aprendam a executar os gestos para palavras e frases simples.

5.4 Engenharia de Requisitos

Para especificar as funcionalidades de “*Os primeiros gestos*” elaborei um documento com o levantamento de requisitos. Como complemento desde documento foram desenhados alguns diagramas de Casos de Uso de forma a elucidar os requisitos e ajudar na análise do problema. Recorri ao *software Sybase PowerDesigner* para criar estes artefactos por ser uma aplicação muito completa em termos de funcionalidades virada para a documentação.

5.4.1 Levantamento de Requisitos

Requisitos funcionais

O sistema proposto terá de ter duas componentes. Uma componente será o *front-end*, ou aplicação principal, que permite consultar palavras e ver a tradução do seu significado em LGP, entre outras funcionalidades. A segunda componente será o *Backoffice* destinada a utilizadores com perfil de Administração para gerir os conteúdos da aplicação principal.

Em seguida enumero os requisitos funcionais da aplicação *front-end*:

- Tradução de palavras para gestos em LGP
- Agrupamento de palavras por Tema
- Pesquisa de palavras
- Combinação de gestos em frases em LGP
- Tradução de frases para gestos em LGP
- Visualização de sugestões de frases existentes no sistema
- Modo exercício - adivinhar o gesto para uma determinada palavra
- Permitir a visualização de gestos com qualidade normal ou em alta-definição
- Permitir a visualização do gesto em câmara lenta e/ou em modo *Loop*
- Sistema de perfis de utilizadores

Os requisitos funcionais do *Backoffice* ou Gestor de Conteúdos são:

- Gerir temas, palavras, gestos e frases
- Gerir utilizadores

- Gerir os temas e palavras associados a cada utilizador

Requisitos não funcionais

Os requisitos não funcionais da aplicação front-end são:

- **Usabilidade:** O sistema é focado para crianças e convém ter um interface simples, apelativo e intuitivo que permita ao utilizador chegar ao seu objetivo de forma eficaz e satisfatória.
- **Extensibilidade:** O sistema deve permitir adicionar novas entradas relativo a temas, palavras, gestos e frases sem ter que recompilar a aplicação.

5.4.2 Análise de Requisitos

Uma técnica importante muito comum no desenvolvimento de sistemas de informação é a identificação dos Casos de Uso do sistema, de modo a facilitar a análise e especificação de requisitos. A descrição formal dos diagramas de caso de uso pode ser consultada nos Anexos *Anexo 5* e *Anexo 6*

Uma vez que o sistema é formado por dois componentes, apresentam-se os diagramas de Casos de Uso tanto para o *front-end* como para o *backoffice*, acompanhado da identificação e descrição dos atores do sistema.

Caso de Uso – Autenticação no sistema

Utilizador – pessoa que utiliza a aplicação *front-end*.

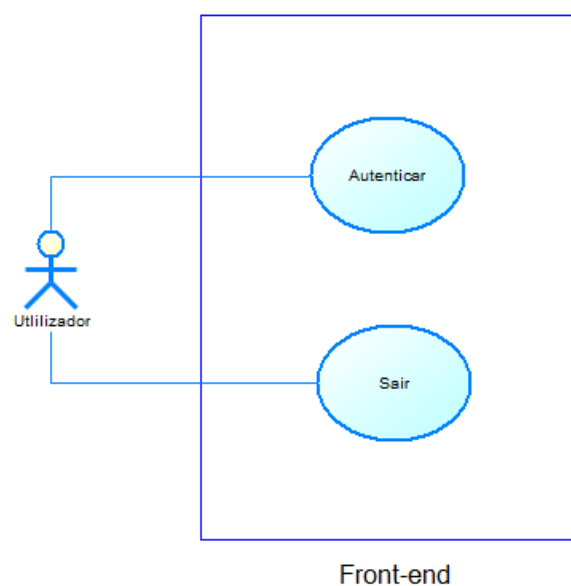


Figura 64 - Caso de uso da Autenticação no sistema

O utilizador pode usar todas as funcionalidades do *front-end* sem se autenticar (*Figura 64*). Contudo, foi planeado estrategicamente a possibilidade de autenticação com vista à evolução futura do sistema. O exemplo que se implementou para demonstrar o potencial, foi a possibilidade de um utilizador ter acesso a diferentes gestos e/ou temas diferentes de outro utilizador.

Nota: A funcionalidade de registo não foi contemplada propositadamente. É necessário primeiro definir inequivocamente o contexto e a finalidade da autenticação. Existem várias possibilidades de uso: filtrar utilizadores comuns de utilizadores *premium*, ou seja, que pagaram para obter mais conteúdos; ser usada no contexto aluno-professor, onde os alunos são orientados por um professor que define como objeto de estudo, um conjunto de gestos. Portanto, após a definição clara do contexto da autenticação, procede-se à criação de um *interface* de registo adequado ao seu objetivo.

Caso de Uso – Traduzir e Adivinhar gesto de Palavras

Utilizador – pessoa que utiliza a aplicação *front-end*.

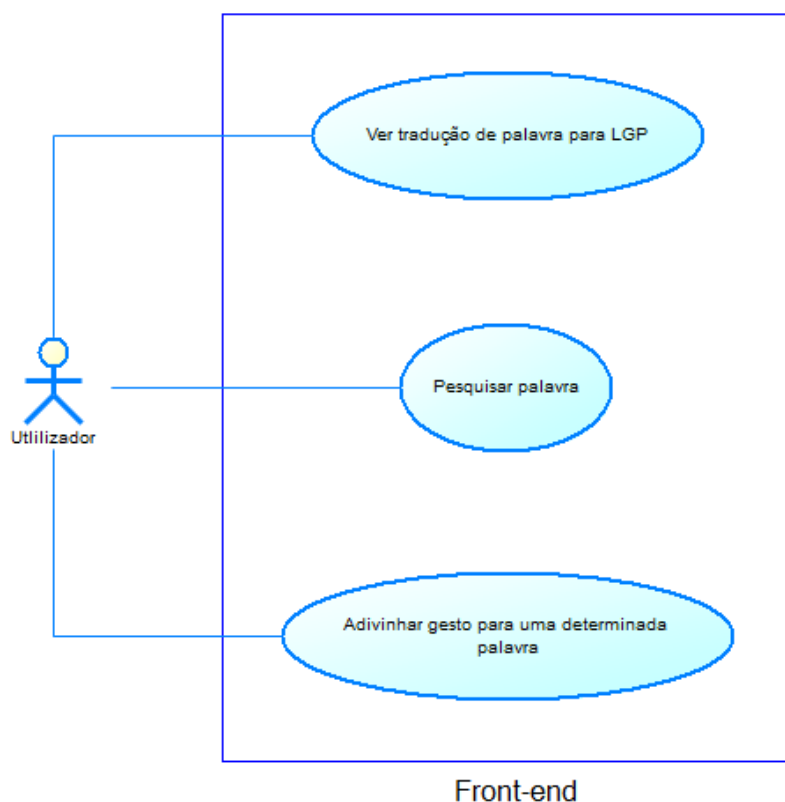


Figura 65 - Caso de uso para Traduzir e Adivinhar gestos de palavras

Este caso de uso (*Figura 65*) representa as ações possíveis do *front-end* referentes a uma palavra ou gesto. Relativo a cada palavra do sistema, o utilizador pode ver a sua tradução no gesto respetivo, efetuar uma pesquisa e por último, pode entrar no modo exercício e adivinhar o gesto que corresponde a cada palavra no sistema.

Caso de Uso – Compositor de Frases

Utilizador – pessoa que utiliza a aplicação *front-end*.

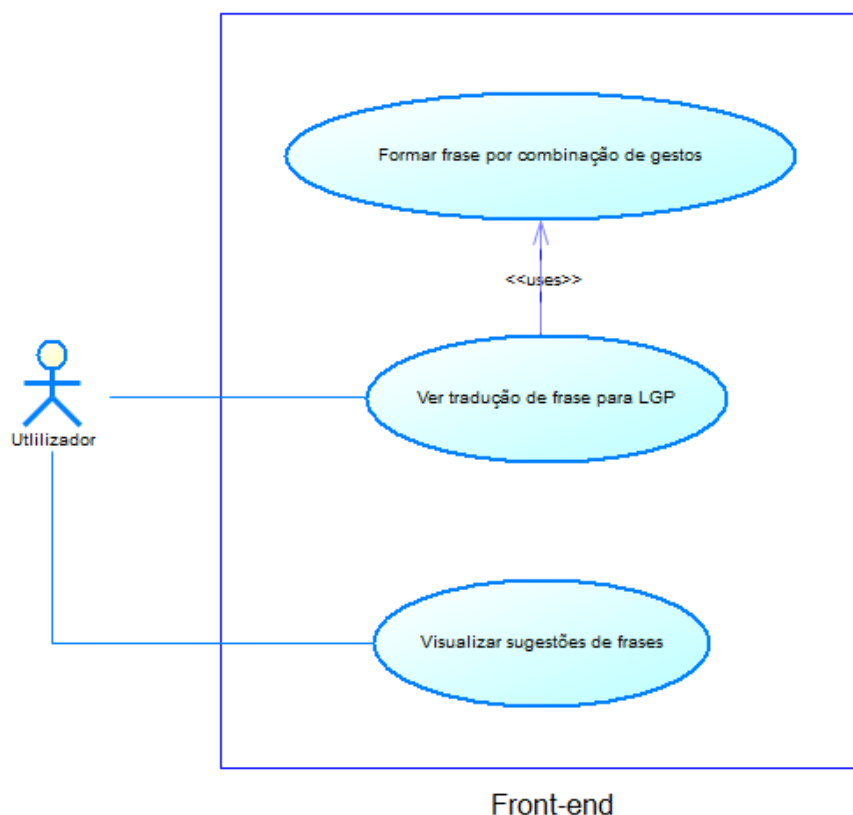


Figura 66 - Caso de uso para Compositor de Frases

Neste caso de uso (*Figura 66*), foram representadas as ações disponíveis ao utilizador relativamente a frases em LGP. O compositor de frases permite a combinação de várias palavras de modo a formar uma frase em LGP. Depois do utilizador combinar as palavras, pode visualizar a tradução da frase em gestos de LGP.

O sistema também permite mostrar várias frases aleatórias existentes na aplicação com o intuito de servir de modelo à composição de frases.

Caso de Uso – *Backoffice*

Administrador – pessoa com privilégios que utiliza a aplicação *Backoffice*.

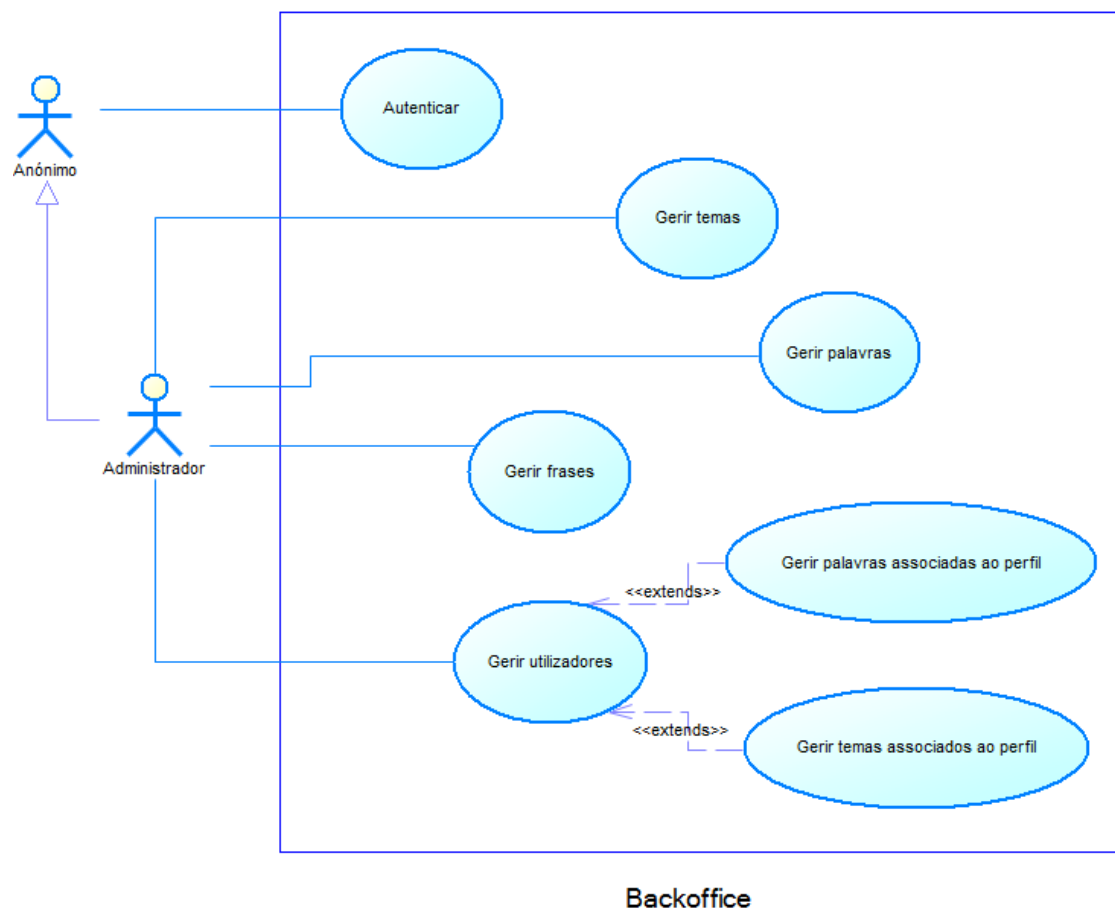


Figura 67 - Caso de uso para *Backoffice*

O *Backoffice* é uma aplicação de suporte ao *front-end* cuja função é permitir a gestão de conteúdos do *front-end*. Por essa razão, no caso de uso da Figura 67 definiram-se as várias funções de *CRUD* (*create, read, update, delete*) que permitem gerir os conteúdos do *front-end*, nomeadamente os temas, palavras, gestos e frases.

É no *Backoffice* que também se gere os utilizadores e se define quais as palavras e temas que cada utilizador pode ter acesso.

5.5 Desenho do Sistema

Esta secção começa por apresentar uma vista geral do sistema através da definição da arquitetura utilizada. Nesta vista faz-se a definição dos componentes físicos do sistema e as suas relações.

Em seguida, apresento o modelo ou a estrutura de dados que serviu de base à implementação da persistência de dados do sistema. Apresenta-se um diagrama de classes onde estão modeladas as entidades e as suas relações da aplicação *front-end*. Também foram identificados os ficheiros *javascript* implementados no lado do *browser*.

Por fim, descrevo o processo e os resultados do método utilizado para o *design* e conceção das *interfaces*.

5.5.1 Arquitetura da aplicação

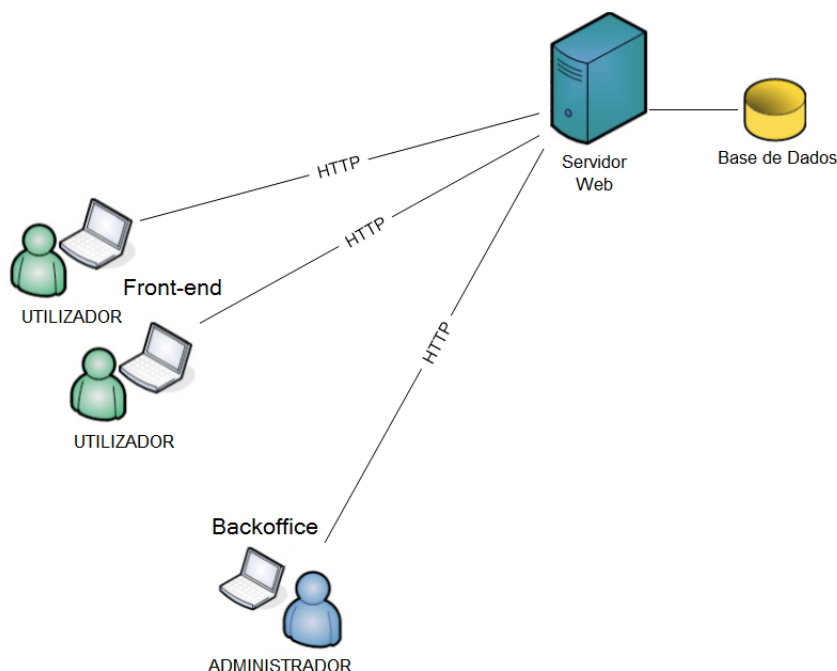


Figura 68 – Arquitetura da aplicação

Front-end e Backoffice

Estes são os componentes que são executados através de um *browser* e fazem a *interface* ao utilizador (Figura 68). O *front-end* é a aplicação *Web* principal enquanto o *Backoffice* ou Gestor de Conteúdos é uma aplicação *Web* de suporte. Cada aplicação tem como função capturar os *inputs* do utilizador, enviar pedidos ao servidor *Web* e apresentar a informação recebida.

Tecnologias Usadas

Usei *HTML 5*, *CSS3*, *JavaScript*, *AJAX*

Servidor Web

O servidor *Web* é responsável pelo processamento dos pedidos da aplicação *Web*. Este componente implementa a lógica de negócio, e sua função é receber o pedido feito pela aplicação *Web*, processá-lo e enviar a resposta. Para concretizar esta tarefa faz pedidos à base de dados.

Tecnologias Usadas

ASP.NET, *C#*

Base de Dados

Este é o componente que permite a persistência de dados do sistema. Realiza todas as tarefas de leitura e escrita dos dados através de um SGBD (Sistema de Gestor de Base de Dados).

Tecnologias Usadas

Microsoft SQL Server

5.5.2 Estrutura de Dados

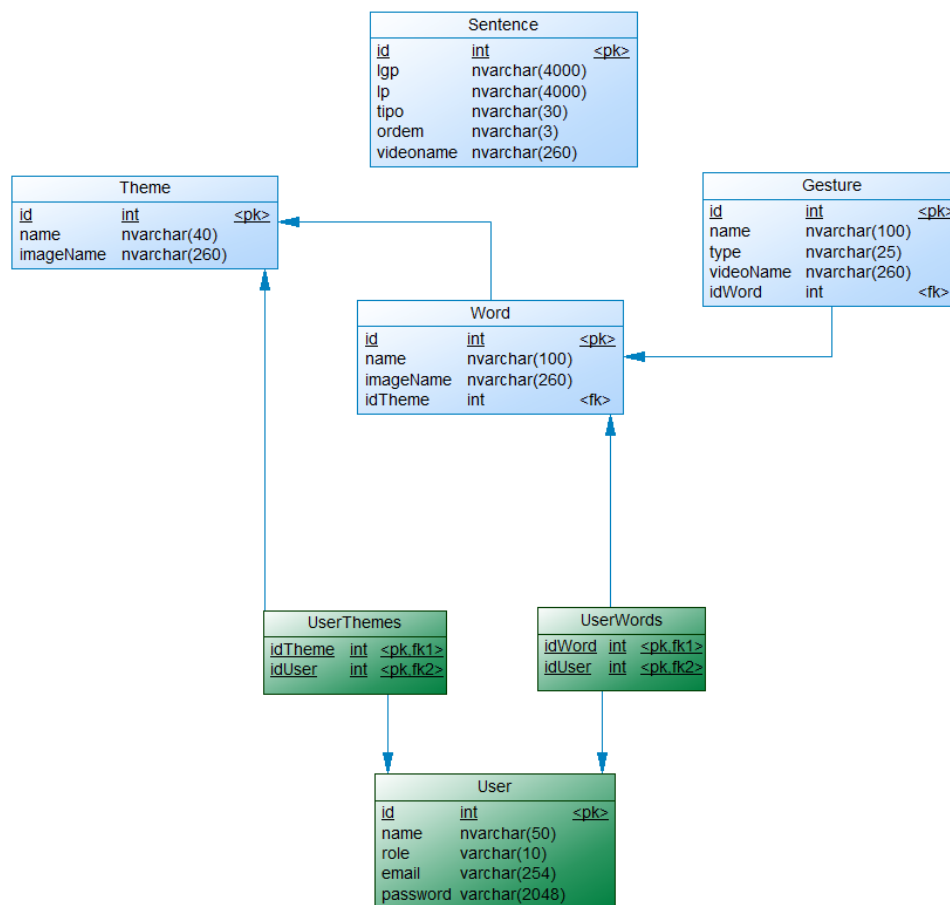


Figura 69 – Modelo de Dados do sistema

Na *Figura 69* representou-se com a mesma cor, as tabelas que têm uma função em comum. A azul estão as tabelas que representam os conteúdos, enquanto a verde estão as tabelas que representam os dados do utilizador. As tabelas a azul *Sentence*, *Theme*, *Gesture* e *Word* representam os conteúdos da aplicação. Cada *palavra* está associada a um *tema*, e cada *gesto* está associado a uma *palavra*.

As tabelas a verde representam os dados do utilizador e a tabela *UserThemes* e *UserWords* existem para implementar a lógica muitos para muitos entre o utilizador e temas/palavras respetivamente.

5.5.3 Diagrama de classes

No diagrama da *Figura 70*, usou-se tal como na *Figura 69*, a mesma cor nas tabelas que tenham uma função em comum. Neste diagrama estão representadas as classes utilizadas no lado do servidor relativas ao *front-end*:

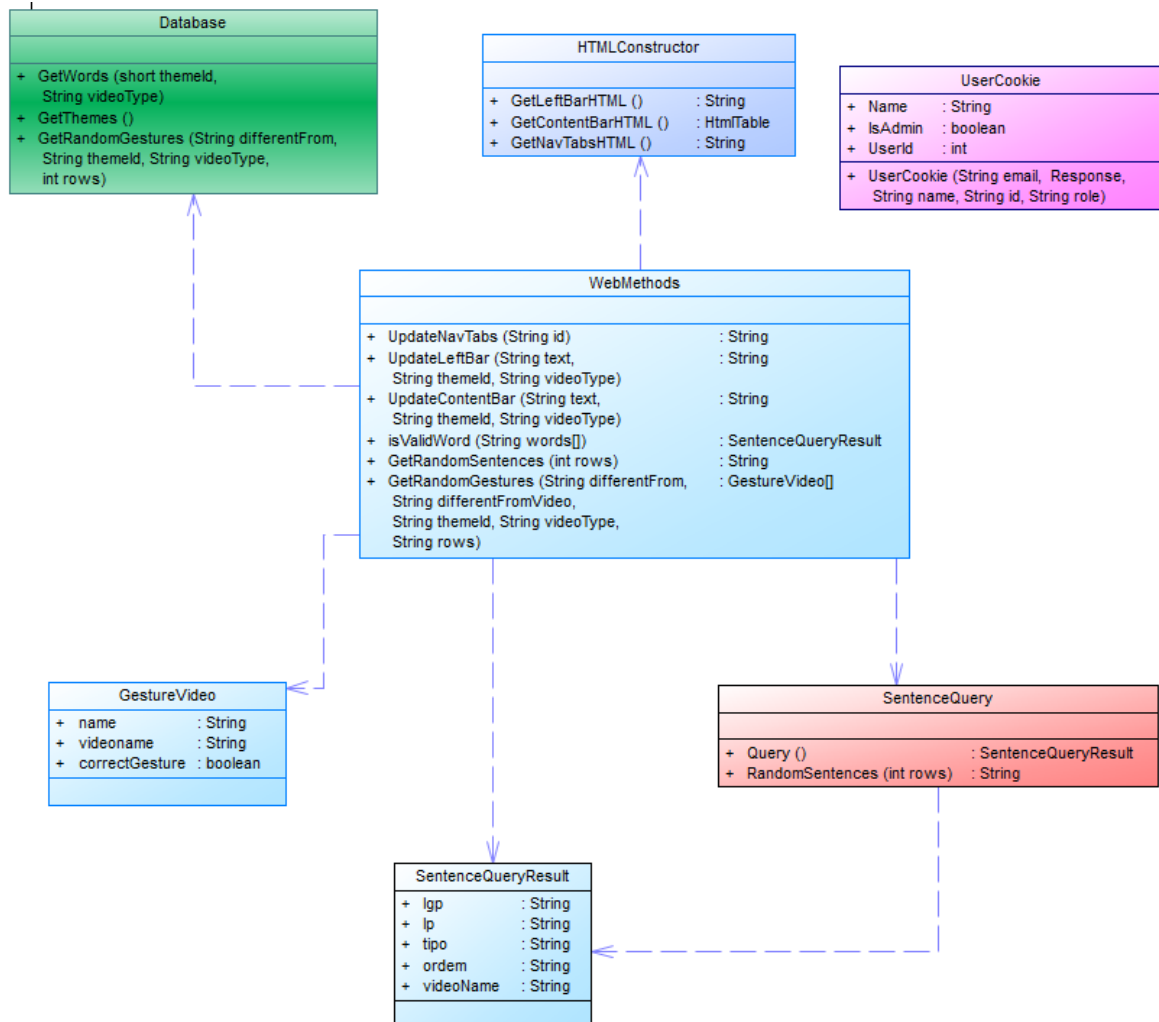


Figura 70 – Diagrama de classes do lado do servidor (*front-end*)

GestureVideo e *SentenceQueryResult* – representam as classes de objetos que são enviados do servidor para o *browser*.

WebMethods é a classe que implementa os métodos *AJAX* chamados por *javascript* no lado do *browser*. É esta classe que faz de *interface* entre os métodos do cliente e a sua implementação no lado do servidor.

A classe *HTMLConstructor* tem como função converter os dados fornecidos pela base de dados em código *HTML* apresentável.

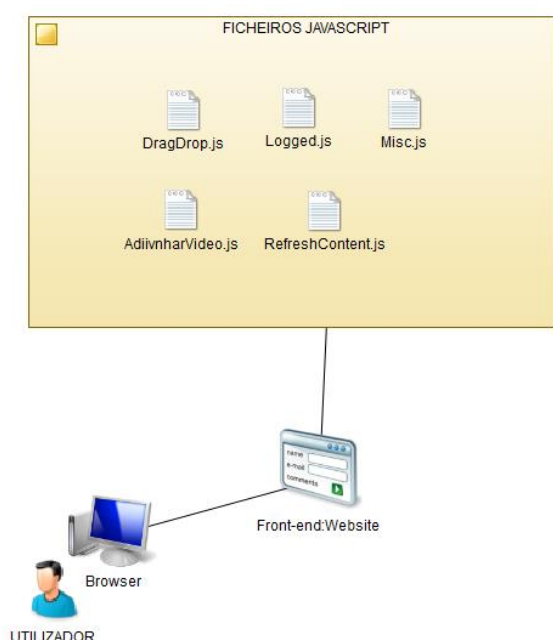
A classe *Database* trata de fazer pedidos à base de dados e retornar essa informação.

A classe *SentenceQuery* é a classe que gere os pedidos do *front-end* relativamente a frases, designadamente, retorna frases aleatórias (visualizar Sugestões de frases) ou responde aos pedidos do componente Compositor de Frases.

A classe *UserCookie* trata de gerir os *cookies* utilizados na aplicação, designadamente tem a função de criar ou ler um *cookie*.

Modelo alto-nível dos ficheiros *javascript* usados no *browser*

Em *javascript* não existe o conceito de classe nativa na linguagem, no entanto, a *Figura 71* mostra os ficheiros de código *javascript* implementados no lado do *browser*



Legenda

DragDrop.js – responsável por toda a lógica de manipulação de objetos arrastáveis.

Logged.js – responsável por gerir o *interface* conforme o utilizador esteja autenticado ou não, e por gerir o *cookie*.

Misc.js – nenhuma função em específico, código comum a várias páginas.

AdivinharVideo.js – trata da interação do *interface* e da lógica de pedidos ao servidor relativos ao componente Adivinhar gesto.

Figura 71 – Ficheiros *javascript* usados no lado do cliente (*browser*)

RefreshContent.js – trata da comunicação com o servidor e das alterações do *interface* relativas ao menu *tabs* e à informação das palavras apresentadas na lista de palavras, ou no centro da *interface*.

5.5.4 Design e Conceção de Interfaces

O *design* e a conceção das *interfaces* seguiram o modelo de desenvolvimento centrado no utilizador. Foram abordadas algumas técnicas de Interação Pessoa-Máquina que passo a descrever em seguida.

O primeiro passo consistiu na definição de *personas* e cenários. O segundo passo foi a elaboração de protótipos. Primeiro desenvolvi iterativamente protótipos de baixa-definição. Em seguida, elaborei os protótipos de alta-definição. Ao longo do processo elaborei vários *designs* alternativos até chegar ao interface mais adequado. O desenho de interfaces incidiu sobre a aplicação principal, o *front-end*. A aplicação *Backoffice* é uma aplicação de suporte à aplicação principal pelo que não foi especificado como requisito a usabilidade. As aplicações como o *Backoffice*, que gerem conteúdos, têm o mesmo objetivo, logo têm uma estrutura similar e o seu tipo de utilizador (administrador) tem um perfil mais experiente que o utilizador comum.

Compreensão dos Utilizadores

A base para um bom *design* de *interfaces* passa por compreender os utilizadores e as suas tarefas de modo a obter um processo de desenvolvimento centrado no utilizador.

Conhecer os utilizadores, quem são, o que fazem, como fazem e quando fazem, foi o objetivo da definição de *Personas* e Cenários apresentados em seguida.

Personas

Personagem 1

Daniel é um miúdo de 6 anos e é surdo. Ele gosta muito de brincar com os seus *Gormities* e o herói preferido dele é o Super-Homem, porque este é diferente dos outros tal como ele.

Tem um computador com Internet que usa muitas vezes para jogar.

Ao contrário dos seus pais que são ouvintes, Daniel é surdo de nascença, no entanto, fala língua gestual portuguesa que aprendeu espontaneamente em contacto com pessoas proficientes na língua. Andará no 1º ano de escola onde está a aprender língua portuguesa, a sua segunda língua.

É um miúdo um pouco fechado e o ensino da sua escola é bilingue (Língua Portuguesa e LGP).

Personagem 2

Manuel tem 29 anos de idade e vive em Lisboa. É um jovem alegre e bem-humorado. Tirou uma licenciatura em Engenharia Civil, mas até agora não encontrou emprego na área. Trabalha como comercial numa loja de vender telemóveis.

Gosta muito de ler e ir ao cinema. Joga todas as semanas no Euro milhões mas o único prémio que ganhou foram 10 euros. Um dos seus sonhos é comprar um iate e ir viver para a Austrália.

Não se dá muito bem com computadores, pois só teve o seu primeiro computador aos 18 anos.

Manuel foi tio há pouco tempo de um menino surdo.

Cenários

Cenário 1

Manuel entra na aplicação. Seleciona a opção de Dicionário. O sistema apresenta um ecrã com vários temas. Escolhe o tema dos animais. O sistema retorna uma lista de vários animais em que cada animal é acompanhado de uma imagem. Manuel seleciona a imagem do gato. O sistema abre um vídeo onde se vê a animação de um humano virtual a executar o gesto em Língua Gestual Portuguesa para a palavra gato.

Cenário 2

Daniel entra na aplicação. Seleciona a opção Adivinhar. O sistema apresenta um ecrã com vários temas. Daniel escolhe o tema das cores. O sistema retorna uma lista de várias cores em que cada cor é acompanhada de uma imagem. Ele seleciona a cor roxa. O sistema apresenta várias animações de gestos. Daniel escolhe uma delas, mas não corresponde à cor certa. Sistema avisa utilizador que gesto não corresponde à palavra e diz que aquele gesto é para a cor vermelha. Daniel volta a tentar, e seleciona outro gesto. O sistema avisa utilizador que está certo e volta ao menu da listagem de cores.

Protótipos

Nesta secção apresento os vários protótipos construídos durante o *design de interfaces* tanto de baixa fidelidade (BF) em papel, como os de alta fidelidade (AF). É descrito o método de conceção dos protótipos e a evolução do seu *design*.

Protótipos de baixa fidelidade

No esboço do lado esq. da *Figura 72* está representado o *interface* inicial da aplicação que serve de entrada ao sistema e define à partida, o modo que o utilizador pretende utilizar a aplicação. A aplicação pode ser usada como dicionário, ou como teste aos conhecimentos (modo exercício/adivinhar).

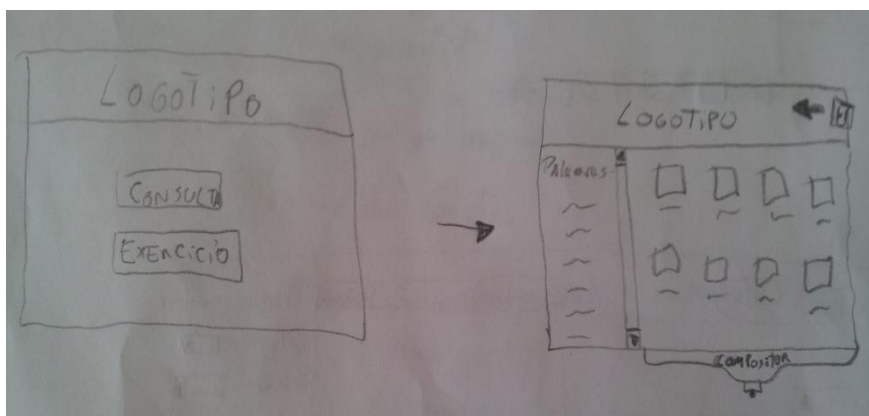


Figura 72 – Protótipo de BF: interface inicial à esq. e interface principal à direita

Após a seleção em *Consulta*, dá-se o seguimento para o interface principal, representado no esboço à direita. Achei que este interface podia ser melhorado, então iterei mais uma vez o seu desenho com os objetivos de adicionar uma barra de *tabs* para escolha de temas, uma *combobox* para alternar entre modo o *Consulta* e *Exercício*, e o menu *Compositor de Frases*. A iteração resultou no esboço à esquerda da Figura 73:

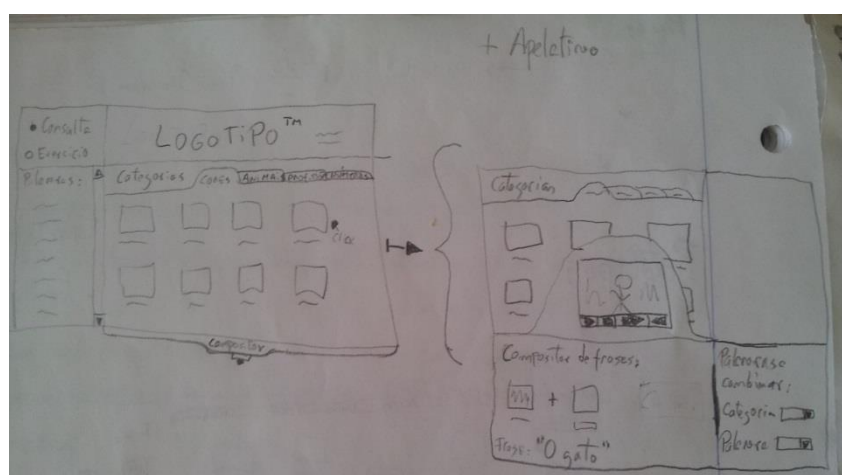


Figura 73 – Protótipo de BF: interface principal iterado à esq. e interface vídeo à direita

Na Figura 73 está representado um *click* numa suposta palavra e a mudança do interface em reação a essa ação. No esboço da direita, vemos o aparecimento do controlo de vídeo a sobrepor-se por cima dos outros elementos. Esta decisão deveu-se à intenção de criar um interface que permita ver um vídeo de uma palavra e ao mesmo tempo ter disponível o máximo número de palavras passíveis de serem seleccionadas. Também se verificou a expansão do menu *Compositor de Frases* que permite combinar palavras através de um menu de escolha de palavras.

O desenho do controlo de vídeo e do *Compositor de Frases* foi iterado várias vezes para chegarem à sua melhor localização espacial no interface como se apresenta na **Figura 74**:

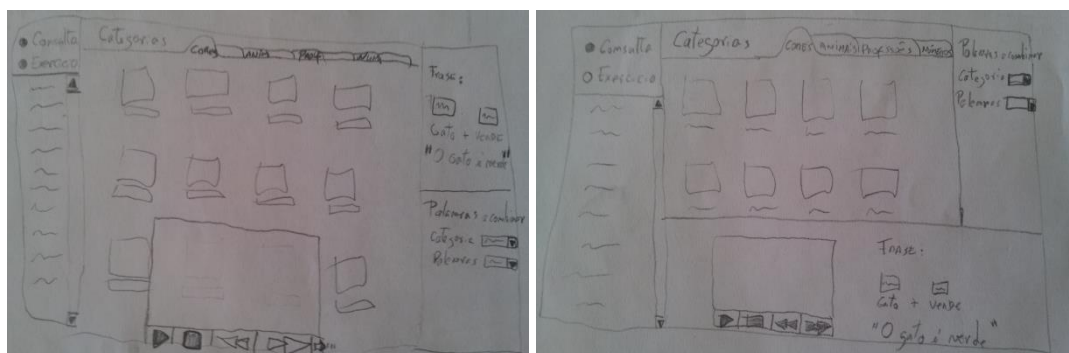


Figura 74 – Protótipo de BF (iteração sobre o interface compositor de frases)

Protótipo de baixa fidelidade final vs. alta fidelidade inicial

Ainda foram feitas mais algumas iterações nos protótipos de baixa fidelidade. A *Figura 75* e a *Figura 76* mostram o passo seguinte a esse processo e a comparação com os protótipos de alta-fidelidade iniciais.

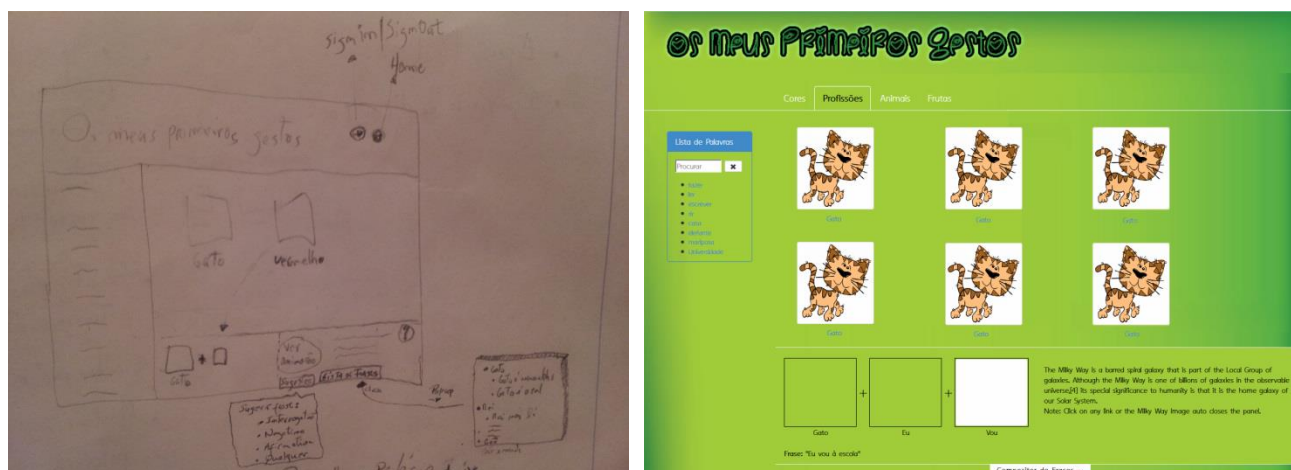


Figura 75 – Interface principal (protótipo de BF à esq. vs. AF à direita)



Figura 76 – Interface para visualizar vídeo (baixa à esq. vs. alta fidelidade à direita)

A *Figura 76* mostra a introdução de uma barra arrastável que define a velocidade do vídeo. Também se verifica a adição de dois botões HD e SD que permitem ver o vídeo em alta-definição ou qualidade *standard* respetivamente, mas que no protótipo de AF inicial ainda não tinham sido desenhados.

Protótipos de alta fidelidade

Até chegar ao *design* mais satisfatório, o *design* de *interfaces* dos protótipos de alta fidelidade implicou várias iterações. Surgiram várias soluções alternativas de desenho pelo que foi escolhido em cada situação, o *design* que achei mais eficaz, coerente e fácil de usar para o utilizador.

Sabendo que um dos públicos-alvo são crianças, procurou-se a criação de *interfaces* interativas e cativantes. Era necessário construir um *interface* colorido e com cores vivas. A experiência do público-alvo com computadores não é muito grande e também de modo a conceber um *interface* intuitivo e fácil de usar, pretendi criar um *interface* simples, mas funcional.

Design gráfico

A imagem de fundo utilizada encontrou-se na Internet. O logotipo foi criado em *PhotoShop* usando as cores da imagem de fundo escolhida. A fonte usada na aplicação foi encontrada na Internet e na *Tabela 4* apresentam-se as fontes encontradas. Decidi escolher por comparação, as duas últimas fontes: a penúltima para o corpo da aplicação e a última para o logotipo.

Fonte	Exemplo
ComingKiddo	Os meus primeiros gestos
GoldPanda	OS MEUS PRIMEIROS GESTOS
KiddySans	Os meus primeiros gestos
CuteLove	OS MEUS PRIMEIROS GESTOS

Tabela 4 – Fontes encontradas para definição do *interface*

Lista de Palavras

A lista de palavras permite ao utilizador ver o conjunto de todas palavras para o tema selecionado. Foi pensado para o utilizador ter uma visão geral de todas as palavras em texto, e caso existam muitas é mais fácil selecionar nesta lista uma palavra do que na secção onde se misturam imagens com palavras. Também foi adicionado uma caixa de texto para a pesquisa de modo a filtrar a quantidade de palavras. A pesquisa funciona à medida que o utilizador escreve os resultados da pesquisa vão sendo apresentados. Isto implementa um *interface* interativo, eficiente

e fácil de usar. Na *Figura 77* mostra-se algumas soluções de *design* alternativas que obtive, e a solução escolhida:



Figura 77 – Evolução do protótipo de AF para a *Lista de Palavras* (evolução em cima, versão final em baixo)

Botões de Opções

Os botões de opções encontram-se no canto superior direito da aplicação e permitem efetuar *login/logout*, ver definições da aplicação e voltar ao *interface* inicial.

As soluções alternativas de *design* que encontrei foram (*Figura 78*):



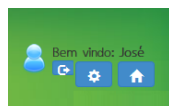


Figura 78 – Evolução do protótipo de AF para *Botões de Opções* (evolução em cima, versão final em baixo)

Compositor de Frases

O compositor de frases permite ao utilizador combinar gestos de modo a formar frases em LGP. Depois de formar uma frase segundo as regras sintáticas da LGP, pode visualizar a tradução em LP da frase que compôs.

Este elemento visual existe na margem inferior da *interface* principal. Por conseguinte, vai “roubar” atenção a outros elementos dessa *interface*. Deste modo, desenhou-se o compositor de frases de forma a que o conteúdo deste componente apenas seja visível quando o utilizador assim o entende. O conteúdo apresenta-se invisível ao utilizador quando entra na página e só se abre quando este clica em cima do separador, mantendo assim o *interface* principal simples e não confuso.

Inicialmente no desenho do protótipo de BF, o compositor de frases tinha *comboboxes* para ir adicionando palavras. No entanto, optei por definir que a construção de frases deve ser feita através do método de *drag and drop*, isto é, o utilizador vai arrastando sequencialmente uma palavra ou imagem para cima de uma caixa dentro do compositor. Desenhou-se a interação desta maneira para implementar um *interface* simples, interativo e divertido (*Figura 79*).

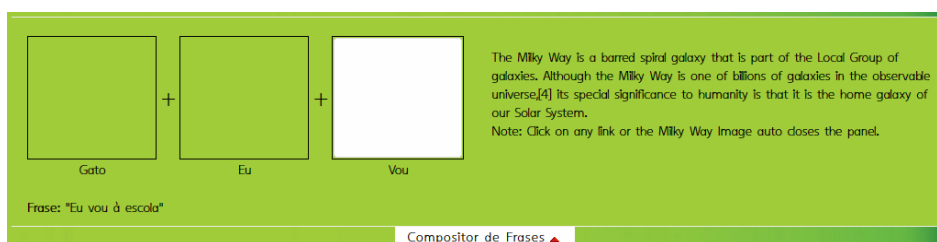


Figura 79 – *Compositor de Frases* (protótipo de AF)

5.6 Implementação

Em seguida, serão descritas as funcionalidades implementadas tanto do *front-end* como para o *backoffice* do ponto de vista técnico. Começo por fazer um apanhado das bibliotecas / *API's* que foram utilizadas para facilitar o desenvolvimento ou necessárias para implementar determinada funcionalidade. Em seguida, descrevem-se as funcionalidades, acompanhadas de algumas *screenshots* dos elementos ou controlos mais importantes. Identificam-se também os problemas encontrados mais relevantes.

5.6.1 Bibliotecas / *API's*

- *Twitter Bootstrap* – coleção de ferramentas para criar *websites*. Contém botões, painéis, ícones, etc., assim como componentes em *javascript*
- *jQuery* - é uma biblioteca em *javascript cross-browser* que tem como função facilitar e aumentar a velocidade de *client-side scripting*. É também usada por outras *API's* descritas nesta secção
- *ASP.NET AJAX* - extensão ao *ASP.NET* utilizada para fazer a comunicação entre cliente-servidor sem *PostBacks*
- *AJAX Control Toolkit* – conjunto de controlos que implementam funcionalidades *AJAX*.

5.6.2 Sistema de Autenticação

A aplicação *front-end* foi desenhada para ser “aberta” ao público. Não é necessário registo nem autenticação para usá-la. No entanto, foi contemplado um sistema de perfis na estrutura implementada. O sistema permite autenticar utilizadores.

Decisões técnicas tomadas

Por um lado, esta funcionalidade tem um papel estratégico, porque constituiu uma base para a implementação de novos requisitos que assentam num perfil de utilizador. Desde requisitos como: criação de base de dados de utilizadores; filtragem de funcionalidades conforme o perfil; entre outros. Portanto, pensei que seria útil criar uma estrutura que permita que o sistema mais tarde evolua noutras direções.

Por outro lado, foi implementado sobre esta funcionalidade a filtragem de conteúdos como temas e palavras, conforme as permissões do utilizador. Por exemplo: um determinado utilizador pode ter temas que outro utilizador não tem.

Interface de Autenticação

Foi implementado o sistema de autenticação nas duas aplicações: *front-end* e *backoffice* (Figura 80).



Figura 80 – Autenticação (*front-end* à esq. e *backoffice* à direita)

O *front-end* permite autenticar tanto utilizadores com perfil administrador como o resto, enquanto o *backoffice* apenas autentica utilizadores com perfil de administrador. A validação da autenticação é feita pela correspondência entre o endereço de *email* e *password* do utilizador, com os dados existentes na base de dados.

Gestão da sessão

Em relação à gestão da sessão, esta é realizada por métodos distintos para o *front-end* e *backoffice*.

O *front-end* utiliza um *cookie*, criado no lado do Servidor (*Listagem 1*) após validar a autenticação, e envia na resposta ao cliente o *cookie* preenchido comos dados relativos ao *username*, *userid* e perfil. A classe responsável pela gestão da sessão é a classe *UserCookie*.

```
HttpCookie cookie = new HttpCookie(FormsAuthentication.FormsCookieName);
cookie.Expires = DateTime.Now.AddHours(8);
cookie.Values["username"] = name;
cookie.Values["userid"] = id;
cookie.Values["role"] = role;
Response.Cookies.Add(cookie);
```

Listagem 1 - Criação de sessão através de *cookie* explícito (*front-end*)

No caso do *Backoffice*, a criação de sessão usa outro método. É utilizado o sistema de *FormsAuthentication* do *ASP.NET*. Após a validação das credenciais por parte do servidor é criado também um *cookie*, contudo, isso é feito de forma transparente para o programador. Como mostra a *Listagem 2*, além de criar a sessão ainda redireciona para o *interface* seguinte:

```
FormsAuthentication.RedirectFromLoginPage(name, true);
```

Listagem 2 – Criação de sessão e redirecionamento por *FormsAuthentication*

Decisões técnicas tomadas

A gestão da sessão apesar de ser distinta nos dois casos, não teve nenhum motivo em especial para ser diferente. Aconteceu porque o segundo método foi aprendido mais tarde após o desenvolvimento do primeiro, e como parecia mais prático optei por essa forma.

5.6.3 Front-end

A *API Twitter Bootstrap* revelou-se muito interessante: é uma *API* simples cujos controlos têm um *look and feel* bastante atrativo, e que foi adotada desde o início da implementação para criar *interfaces* sofisticadas. Esta biblioteca é *cross-browser*, e apesar de não ser ainda um requisito desenvolver para *mobile*, os seus controlos adaptam-se visualmente às medidas de diferentes dispositivos como PC, telemóvel, etc.

Outra *API* usada foi a *ASP.NET AJAX* de modo a criar *interfaces* interativas. Quase 100% dos controlos utilizam *AJAX*. Os métodos usados para comunicar entre cliente-servidor via *AJAX* foram enumerados na *Listagem 3*:

```
string UpdateNavTabs(string id)
string UpdateLeftBar(string text, string themeId, string videoType)
string UpdateContentBar(string text, string themeId, string videoType)

SentenceQueryResult isValidWord(string[] words)
string GetRandomSentences(int rows)

GestureVideo[] GetRandomGestures(string differentFrom, string differentFromVideo, string themeId, string videoType, string rows)
```

Listagem 3 – Lista de *PageMethods* ou métodos *AJAX* utilizados no *front-end*

Pode ser consultada uma descrição mais detalhada acerca de cada método no *Anexo 7*.

As chamadas à camada de dados usaram as funções do *ASP.NET ADO.NET*.

Modo Dicionário

O modo dicionário constitui o *interface* com mais interatividade e funcionalidades da aplicação. Tem funcionalidade de pesquisa, *drag and drop*, *tabs* para mudar de tema, funcionalidade de ver vídeo para palavra ou frase, e compositor de frases.

Comecei por criar a estrutura de espaços ou zonas através de *HTML/CSS*. O *interface* foi separado por 3 grandes *divs*: *logo*, *contentBar* e *leftBar*. *logo* é onde fica o logotipo e os botões de

opções. *leftBar* é o espaço para a lista de palavras e pesquisa, enquanto *contentBar* foi o espaço onde se situa a maioria da interação, designadamente ver vídeo e compor frases (ver *Figura 81*).



Figura 81 – *divs* do layout: *logo*, *contentBar* e *leftBar*

Botões de opções



Figura 82 – Botões de opções: voltar ao *interface* inicial, ver definições, e fazer *login/logout*

A aplicação tem uma configuração (*Figura 82*) que permite alternar entre a visualização de vídeos com humanos, ou então visualização de animações com avatar. Quando usada a configuração de gestos com avatar, caso não haja animação com avatar disponível para uma determinada palavra, é fornecido o vídeo com humanos, caso exista no sistema.

Lista de Palavras

A lista de palavras permite uma pesquisa por gestos (*Figura 83*). Foi feito de modo que basta o utilizador escrever, a pesquisa é feita automaticamente em *background* e os resultados são apresentados ao utilizador sem necessidade de carregar *enter*.

Para isso associou-se o evento *KeyUp* da caixa de texto da pesquisa ao método *SendClick(event)* presente na *Listagem 4* que por sua vez chama métodos AJAX:

```
function SearchClick(event) {  
  
    var themeId = activeTabId  
    var text = document.getElementById('SearchText').value  
  
    PageMethods.UpdateLeftBar(text, themeId, "animation", OnUpdateLeftBarSuccess);  
    PageMethods.UpdateContentBar(text, themeId, "animation", OnUpdateContentBarSuccess);  
}
```

}

Listagem 4 – Código para fazer uma pesquisa (*front-end*)

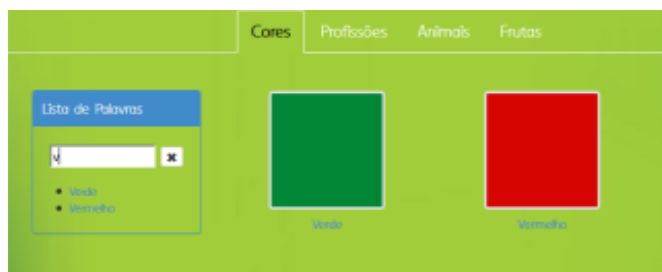


Figura 83 - Lista de palavras

Ver Vídeo

A visualização do vídeo por motivos já justificados na secção dos *Protótipos de baixa fidelidade*, usa um painel que aparece no canto inferior direito. Esse painel usa o componente *AjaxSidePanel*¹². Basicamente é uma *div* criada dinamicamente em *javascript* sempre que se clica numa palavra.

O controlo necessário para mostrar o vídeo foi o *flowplayer*¹³. Esta escolha reside no facto de ser um controlo esteticamente simples, muito configurável e com disponibilização de uma *API* robusta e bem documentada para interagir com o componente. Para fazer o *streaming* de vídeo o componente usa *HTML 5* e caso o *browser* do cliente não suporte, usa *Flash*.

Um dos requisitos da aplicação era ver o vídeo em repetição e conseguir controlar a velocidade do vídeo. Então implementou-se um botão para a funcionalidade *Loop* (*Listagem 5*).

```
/* Makes the Video player loop the animation when user clicks the loop button */  
var loop = true  
function ToggleLoop() {  
  
    if (loop) {  
        document.getElementById("videoFlow").setAttribute("loop", "")  
        flowplayer().play()  
    }  
    else  
        document.getElementById("videoFlow").removeAttribute("loop")  
}
```

¹² *AjaxSidePanel* – controlo disponível em <http://www.dynamicdrive.com/> acedido a 20 de Junho de 2013

¹³ *flowplayer* – controlo disponível em <http://flowplayer.org/> acedido a 20 de Junho de 2013

```
loop = !loop;  
}
```

Listagem 5 - Código para alternar entre vídeo com *loop* activou ou inativo

Também se implementou com recurso a um *slider* do *AJAX Control Toolkit*, um controlo para gerir a velocidade do vídeo, como se verifica na *Figura 84*. Considerei usar um *slider* da biblioteca *jQueryUI*, mas achei esteticamente melhor o do *AJAX Control Toolkit*, e além disso já estava a usar esta biblioteca para outras funcionalidades.

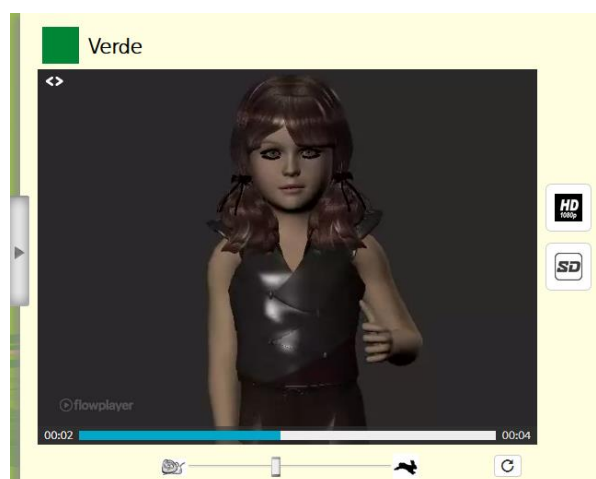


Figura 84 – Ver Vídeo

Compositor de frases

O compositor de frases é o espaço onde o utilizador pode combinar gestos e compor frases com a sintaxe da LGP. Como já foi justificado na secção *Compositor de Frases* do Desenho do Sistema, o compositor não está sempre visível, então foi necessário usar um painel que colapsa o seu conteúdo dinamicamente. Quando o utilizador clica num determinado botão o painel do compositor de frases abre-se e vemos o seu conteúdo, voltando-se a carregar o painel fecha-se. Recorreu-se ao uso do controlo *Dropdownpanel*¹⁴ para implementar esta funcionalidade.

¹⁴ *Dropdownpanel* – controlo disponível em <http://www.dynamicdrive.com/> acessido a 29 de Junho de 2013

Caso o utilizador não saiba a ordem frásica da composição de frases em LGP, pode visualizar sugestões de frases que apresentam uma comparação entre frases em LGP e frases em LP. Para isso o utilizador basta carregar no *hyperlink* chamado “*sugestões aqui*”.

Sugestões de frases

Esta funcionalidade funciona da seguinte forma: quando o utilizador carrega a página, é pedido ao servidor um número de sugestões aleatórias que são guardadas no *browser*. Quando o utilizador pretender ver as sugestões ao carregar no *hyperlink* de sugestões, então são apresentadas as sugestões previamente carregadas.

Após fechar o painel das sugestões são pedidas novas sugestões ao servidor.

Drag-and-drop

Para facilitar a implementação a funcionalidade de *drag-and-drop* existe a biblioteca *jQueryUI*, o suporte nativo do *HTML5* e do *ASP.NET AJAX*. No entanto, foi escolhido o suporte do *ASP.NET AJAX* pois após a análise a um código fonte exemplo, a sua implementação parecia simples e prática.

Implementação do Drag-and-drop

É preciso definir como *DragSource* e *DropTarget* os elementos que queremos trabalhar como elementos arrastáveis e elementos que possuam capacidade de serem alvo de *Drops*. O código responsável por isso está apresentado na *Listagem 6*.

```
<script type="text/javascript">

// Make the "DropTarget" DIV a drop target.
var target = new Custom.UI.DropTargetBehavior
($get('DropTarget1'));
target.initialize();

// Make the "DragSource" DIVs drag sources.
var source = new Custom.UI.DragSourceBehavior
($get('wordContainer<%=WordName %>'), '<%=WordName %>', '<%=ImageName %>');
source.initialize();

</script>
```

Listagem 6 - Código de definição de *DragSource*'s e *DropTarget*'s

Quando um elemento é arrastado e entra numa zona de *Drop*, é chamado uma função *AJAX* através de *PageMethods* para saber se aquele elemento pode ser deixado cair ali ou não. Se o elemento é *dropable*, e o utilizador deixa cair o objeto, então definimos a imagem que corresponde à palavra para imagem de *background* do quadrado onde estamos a fazer *drop*.

Para saber se o elemento é *dropable* ou não, é lido a resposta que o servidor enviou.

Resposta do Servidor

A resposta do servidor é um objeto *SentenceQueryResult*. Esse objeto inclui toda a informação relevante à frase, como a frase em LGP, a frase em LP, o tipo de frase, a ordem sintática e o vídeo da frase. A *Listagem 7* mostra os casos possíveis de resposta do servidor.

```
public SentenceQueryResult Query()
{
    var adapter = new DataSets.Sentence_DataSetTableAdapters.SentenceTableAdapter();
    var dt = adapter.GetSentences(wordsSeparatedBySemiColon);

    if (dt.Rows.Count > 0)
    {
        foreach (DataSets.Sentence_DataSet.SentenceRow row in dt.Rows)
        {
            var words = row["LGP"].ToString().Split(';');

            if ((words.Length - 1) == wordsNumber) // means that the sentence matched the words submitted by the user
            {
                return new SentenceQueryResult(row["LGP"].ToString(), row["LP"].ToString(), row["TIPO"].ToString(),
                    row["ORDEM"].ToString(), row["VIDEONAME"].ToString());
            }
        }
        return new SentenceQueryResult(); //means there is one or more rows with the words user submitted or combination
ords
    }
    return null; //means there is no row with that words or combination of words
}
```

Listagem 7 – Código para processar a ação do utilizador quando tenta arrastar uma palavra para o espaço respetivo do Compositor de Frases

Caso não exista frases no sistema que tenham a combinação das palavras que o utilizador está a tentar construir, o objeto é retornado a *null*. Caso exista, um início de frase com as palavras que o utilizador está a tentar combinar, é retornado só a instância da classe com os seus atributos a *null*. Caso a frase combinada esteja completa o servidor retorna uma instância da classe com todos os seus atributos preenchidos corretamente.

Depois de uma resposta positiva do servidor, ou seja, após o utilizador ter construído uma frase correta e completa, são habilitados o botão para traduzir a frase em LGP e é apresentado a versão em Português da frase como se pode ver na *Figura 85*:



Figura 85 – Compositor de Frases: arrastando uma palavra

Modo Adivinhar / Exercício

O modo adivinhar é a interface que serve para testar os conhecimentos. No entanto, tem muitas funcionalidades em comum da *interface* do modo dicionário, como pesquisa, escolha de tema e visualização de vídeo. Muito código é comum às duas *interfaces*.

Para esta vertente da aplicação ser uma *interface* mais interativa e divertida adicionou-se um elemento visual, um boneco de um menino.

Menino tutor/assistente

Este elemento serve apenas para passar informação para o utilizador. O boneco “fala” através de um balão com a mensagem dentro do próprio balão. Esta funcionalidade foi implementada recorrendo ao *BalloonPopupExtender* da biblioteca *AJAX Control Toolkit*.

Sempre que pretendo enviar uma mensagem ao utilizador, uso o código da *Listagem 8*:

```
$get("BalloonPopupText").innerHTML = "Qual dos gestos corresponde à palavra '" + word + "' ?"
showBalloonPopup(event)
```

Listagem 8 – Código para fazer o menino assistente “falar” (em Adivinhar gesto)

A primeira linha do código é para definir o conteúdo da *div* que contém a mensagem e a segunda é para mostrar o *popup*.

Ver Vídeo

Ao contrário do painel utilizado para mostrar e esconder o vídeo no *Ver Vídeo*, no modo Adivinhar não se usou esse painel. Então recorri a um componente chamado *Animated Collapsible DIV*¹⁵ que serviu para esconder o espaço onde se vai mostrar os gestos. É um componente similar ao usado no *Compositor de Frases*. O painel encontra-se colapsado antes de o utilizador escolher a palavra que quer adivinhar, após a escolha da palavra, o painel abre-se e é apresentado o espaço onde o utilizador visualiza o vídeo e escolhe qual dos gestos apresentados corresponde à palavra.

Usou-se o controlo de vídeo *flowplayer* já referido anteriormente, mas desta vez temos uma *playlist* com vários vídeos.

Playlist

A *playlist* de vídeos é construída dinamicamente. A escolha da palavra implica que um dos vídeos a ser visualizados seja o que corresponde ao gesto. O resto dos vídeos são vídeos escolhidos aleatoriamente, mas que façam parte do mesmo tema que a palavra.

5.6.4 Backoffice

O *Backoffice* foi implementado usando *ASP.NET Dynamic Data*. Esta *framework* permite facilmente criar aplicações *Web* que façam operações CRUD (*create, read, update, delete*) nas tabelas da base de dados. A *framework* automaticamente transforma o conteúdo de uma tabela, em *HTML*, e gera automaticamente o *interface* para gerir os dados de cada tabela.

Esta ferramenta facilitou muito o processo de desenvolvimento do Gestor de Conteúdos. Após a utilização da ferramenta, bastou fazer o *design* gráfico e implementar o sistema de autenticação.

5.6.5 Problemas encontrados

Ver Vídeo – *Slider* de velocidade do vídeo e *flowplayer*

Sempre que se arrastava o *slider* para mudar a velocidade do vídeo aconteciam duas situações. A primeira era que o evento que indicava que o *slider* tinha sido arrastado era duplicado, ou seja, mexia-se o *slider* uma vez, mas o evento era despoletado duas vezes. A segunda situação era que ao mexer um centímetro o *slider*, eram despoletados dezenas de eventos e a atribuição direta da velocidade do vídeo no *flowplayer* gerava um comportamento muito estranho no vídeo.

A solução encontrada foi criar um algoritmo (*Listagem 9*) que basicamente armazenava numa *queue* os eventos que o *slider* despoletava, e só passado um determinado tempo após o movimento inicial do *slider*, é que se definia no *flowplayer* a velocidade de vídeo com o último valor na *queue* de eventos.

¹⁵*Animated Collapsible DIV* – controlo em <http://www.dynamicdrive.com/> acedido a 20 de Junho de 2013

```

/* Catch 'change' event of the Slider (speed slider) - and set the speed of the animation. however doesn't set
immediately, it waits some time first using a timer */

$addHandler($get("txtSpeed"), "change", SliderChanged);

var speedTimes = new Array();
var timerOn = false
/* Update the speed of the animation defined by the user on the slider */
function SliderChanged(x) {
    var slideValue = $get("SpeedBoundControl").innerHTML

    if (timerOn == false) {
        timerOn = true
        //start timer
        window.setTimeout(ChangeVideoSpeed, 200)
    }
    speedTimes.push(slideValue) //add to array
}

function ChangeVideoSpeed() {
    var last = speedTimes.pop();
    speedTimes = [];

    flowplayer().speed(last);
    timerOn = false
}

```

Listagem 9 – Algoritmo para atribuir o valor do *slider* à velocidade de vídeo do *flowplayer*, evitando o comportamento estranho da atribuição direta entre o valor do *slider* e vídeo

Compositor de frases – incompatibilidade entre *Drag and Drop* e Ver Vídeo

O processo *Drag and Drop* implica detetar o *click* do utilizador para iniciar o processo. No entanto, o *Ver Vídeo* implica também que haja um *click* em cima do mesmo elemento que podemos fazer *Drag*. Portanto, isto gerou um problema que apenas uma das funcionalidades podia funcionar ao mesmo tempo.

A solução encontrada foi alterar o evento usado pelo *Drag and Drop* de *mouseDown* para *mouseMove*. Assim sempre que se mexia o *mouse* por cima da palavra, era iniciado o processo de *Drag*. Para evitar isso, detetamos se nessa situação o botão esquerdo do rato estaria a ser pressionado. Se estivesse, então entrava em modo *Drag and Drop*, se não, não entrava.

5.7 Resultados

Algumas *screenshots* do sistema tanto para o *front-end* como para o *backoffice* encontram-se no Anexo 8.

5.8 Sumário

O capítulo descreve exhaustivamente todo o processo de desenvolvimento de uma prova de conceito, designada por “*Os meus primeiros gestos*”, que consiste numa aplicação *Web* com o objetivo de introduzir conceitos básicos de LGP, desenhada principalmente para crianças.

Capítulo 6

Avaliação de “*Os meus primeiros gestos*”

Este capítulo tem como objetivo apresentar os resultados da avaliação com utilizadores realizada à aplicação *Os meus primeiros gestos* em particular ao seu *front-end*.

6.1 Objetivo

A avaliação da aplicação teve três objetivos fundamentais:

- Teste à usabilidade do sistema
- Teste à qualidade de animação (avatar e execução dos gestos)
- Teste à utilidade de determinadas funcionalidades

O teste com utilizadores dividiu-se em três partes.

A primeira parte pretende avaliar **a componente de dicionário**, através do teste à tradução de palavras, a experiência de visualização dos gestos e composição de frases.

A segunda parte pretende testar **o modo adivinhar / exercício**, a componente mais didática da aplicação.

A terceira parte consistiu na recolha da opinião geral sobre a aplicação e a sua utilidade prática, e ainda no levantamento de novas funcionalidades a implementar.

6.2 Participantes

Participaram 3 pessoas, duas surdas e uma ouvinte proficiente em LGP. O conjunto era composto por uma mulher com 25 anos, um homem com 34 e outro com 37.

6.3 Métricas

As métricas para avaliação basearam-se em questionários que medem a usabilidade da aplicação através da execução de tarefas delegadas a cada participante. As tarefas foram

desenhadas de modo a que as funcionalidades da aplicação fossem todas exploradas. Também se observou e registou-se notas sobre o comportamento dos utilizadores durante o uso da aplicação.

Os questionários foram compostos por várias perguntas na maioria usou-se a escala *Likert* de 5 níveis, noutras a escala *Likert* de 3 níveis, e as restantes perguntas eram de respostas abertas.

Os questionários distribuídos estão representados nas tabelas 5, 6, 7:

Pergunta	Escala
Facilidade de execução de tarefas	1- Pouco Fácil 5- Muito Fácil
Utilidade do compositor de frases para aprender LGP	1- Pouco Útil 5- Muito Útil
Facilidade em combinar gestos	1- Pouco Fácil 5- Muito Fácil
Vale a pena existir vídeo HD e normal?	Sim / Não / Talvez
Tamanho da janela de vídeo é adequado?	Pequeno/Grande/Bom

Tabela 5 – Questionário do teste ao Dicionário

Pergunta	Escala
Facilidade na execução de tarefas	1- Pouco Fácil 5- Muito Fácil
Facilidade no uso da pesquisa	1- Pouco Fácil 5- Muito Fácil
Modo Adivinhar interessante?	1- Pouco Interessante 5- Muito Interessante
Modo Adivinhar divertido?	1- Pouco Divertido 5- Muito Divertido
Necessário ver vídeo em câmara lenta quando se tenta adivinhar?	Sim / Não
Opção <i>Full screen</i> é importante?	Sim / Secundário / Não vale a pena

Tabela 6 – Questionário do teste ao modo Adivinhar

Pergunta	Escala
Utilidade da aplicação	1- Pouco Útil 5- Muito Útil
Experiência de utilização / facilidade	1- Muito Má 5- Muito Boa
O avatar é interessante?	1- Nada 5- Muito
Visual atrativo?	1- Nada 5- Muito
Qualidade dos gestos	1- Muito Mau 5- Muito Bom
Se fosse adicionado expressão facial como achas que ficaria a qualidade dos gestos?	1- Muito Mau 5- Muito Bom

Tabela 7 – Questionário Geral

6.4 Método

Na primeira parte o utilizador seguiu o primeiro guião, cujas tarefas testavam as funcionalidades do dicionário, e que pode ser consultado no Anexo 9. Enquanto na segunda parte, o segundo guião (**Anexo 11**) tinha tarefas de forma a testar a vertente de adivinhar palavras da aplicação. Na terceira parte não houve guião, apenas um questionário. O participante realizava tarefa a tarefa à medida que ia lendo o guião. No fim da leitura de cada guião, o participante respondia a um questionário. Finalmente, após ter testado tudo, respondeu a um questionário com questões gerais sobre a aplicação.

Conforme os participantes iam realizando os testes, fui fazendo perguntas quando me apercebia de dificuldades para perceber o que poderia ser melhorado.

Questionário

Os questionários usados na avaliação podem ser consultados no *Anexo 10* para o primeiro questionário, no *Anexo 12* para o segundo e no *Anexo 13* para o terceiro.

6.5 Resultados

Nesta secção apresentam-se os resultados dos questionários, as observações feitas ao longo do teste, assim como também a análise dos anteriores.

6.5.1 Primeiro questionário – teste ao Dicionário

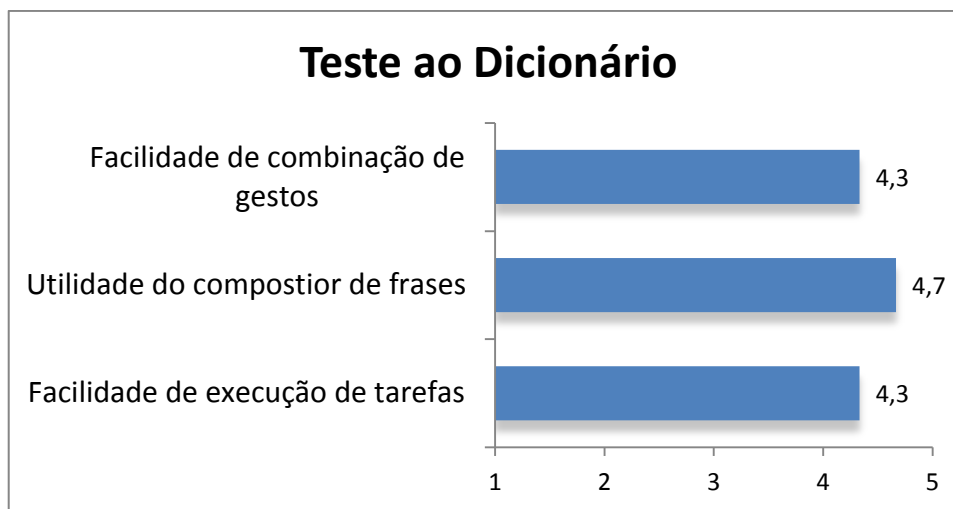


Figura 86 – Gráfico da média de respostas ao teste do Dicionário

O teste às funcionalidades gerais do modo Dicionário revela bons resultados uma vez que todos os valores do gráfico (*Figura 86*) estão acima de 4,3.

Facilidade de execução de tarefas – este indicador representa que no modo geral a vertente de Dicionário da aplicação tem boa usabilidade (4,3), porém, este valor ainda pode, e deve ser melhorado. Uma abordagem possível é a melhoria da usabilidade do *Compositor de Frases*, que por conseguinte, aumenta a usabilidade do dicionário em geral. Além disso vai de encontro com as observações realizadas a um dos participantes.

Um dos surdos testados não conseguiu compreender como se iniciava o processo de combinar gestos, isto é, não compreendeu imediatamente que tinha de se arrastar as palavras. Depois de perceber, achou muito fácil o processo de arrastar, porém, esta situação levantou a necessidade de que é preciso melhorar os indicadores informativos (textuais e principalmente visuais) de como compor frases. O próprio surdo apontou uma das causas para esta situação, isso está descrito em nota de rodapé e será usado como “*guideline*” a futuros desenvolvimentos¹⁶.

Foi quase unânime por parte dos participantes que o *Compositor de Frases* é muito útil no contexto da aprendizagem de LGP. Isto é um indicador que deve se apostar nesta ferramenta, e que a sua evolução deve ser uma prioridade.

Os indicadores apresentados revelam que esta componente da aplicação (dicionário) foi bem desenhada e implementada. Deduzo que o uso extensivo da tecnologia AJAX para atualizar o

¹⁶ O surdo ao início não compreendeu que tinha de arrastar a palavra para compor a frase, apesar da informação textual existente de propósito para ajudar iniciantes. Depois de discutir sobre isto com ele, o surdo disse que os surdos não ligam a texto, e focam-se principalmente nas muito mais nas imagens do que um ouvinte.

conteúdo do *interface*, juntamente com o *javascript* desenvolvido para a interação, contribuiu para este resultado.

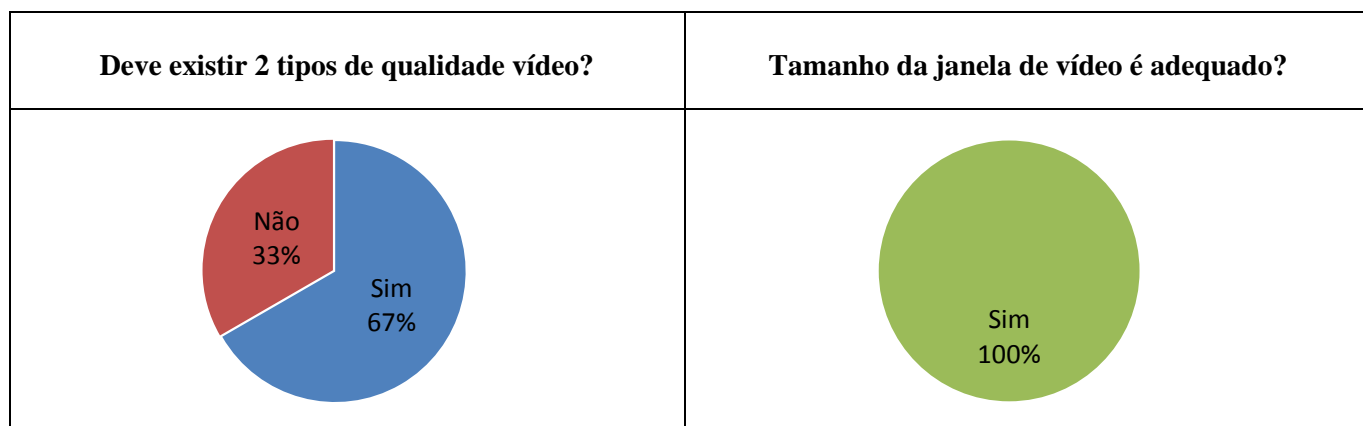


Figura 87 – Gráfico em círculo de respostas do primeiro questionário

Estas perguntas (*Figura 87*) foram feitas para validar se a funcionalidade de ver vídeo em alta-definição tinha utilidade ou não, e se a dimensão da janela de vídeo, uma vez que os surdos são muitos visuais, também estaria adequada. Pelos resultados nos gráficos foram decisões bem tomadas.

6.5.2 Segundo questionário – teste ao modo Adivinhar

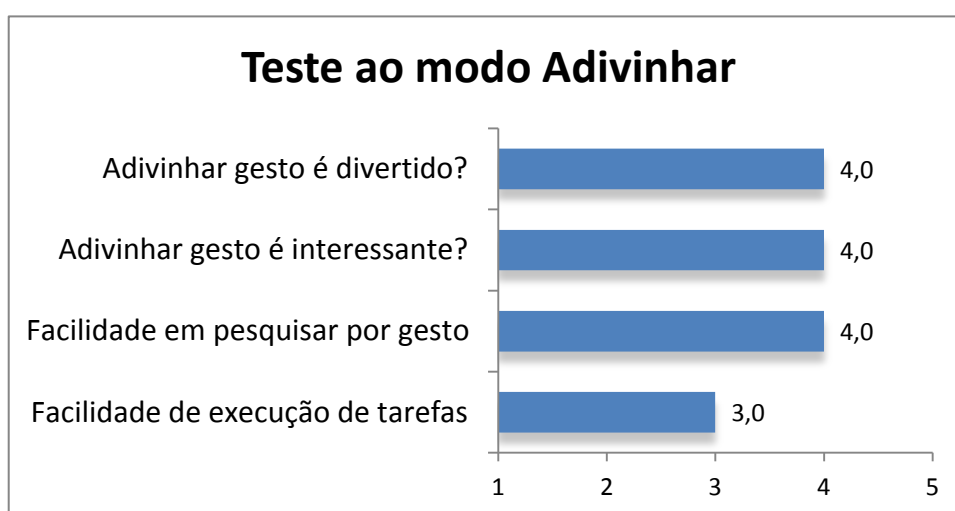


Figura 88 – Gráfico da média das respostas ao questionário do modo Adivinhar

Ao contrário da componente Dicionário, a componente Adivinhar (*Figura 88*) gesto tem apenas um nível satisfatório (3 pontos) na *Facilidade de execução de tarefas*. Esta situação leva que o desenvolvimento prioritário futuro passe a ser a resolução dos problemas identificados neste teste. No entanto, a avaliação de outros fatores que contribuem para a usabilidade do sistema como ser interessante/divertido tiveram um bom *feedback*. Pela observação e discussão com os

participantes, estes referiram que é necessário melhorar o *design* de alguns botões porque confundem o utilizador, nomeadamente, os botões que marca a resposta correta ou errada. Ainda sugeriram acrescentar um pouco mais de interatividade como adicionar um elemento surpresa sendo o próprio sistema a escolher o gesto para adivinhar.

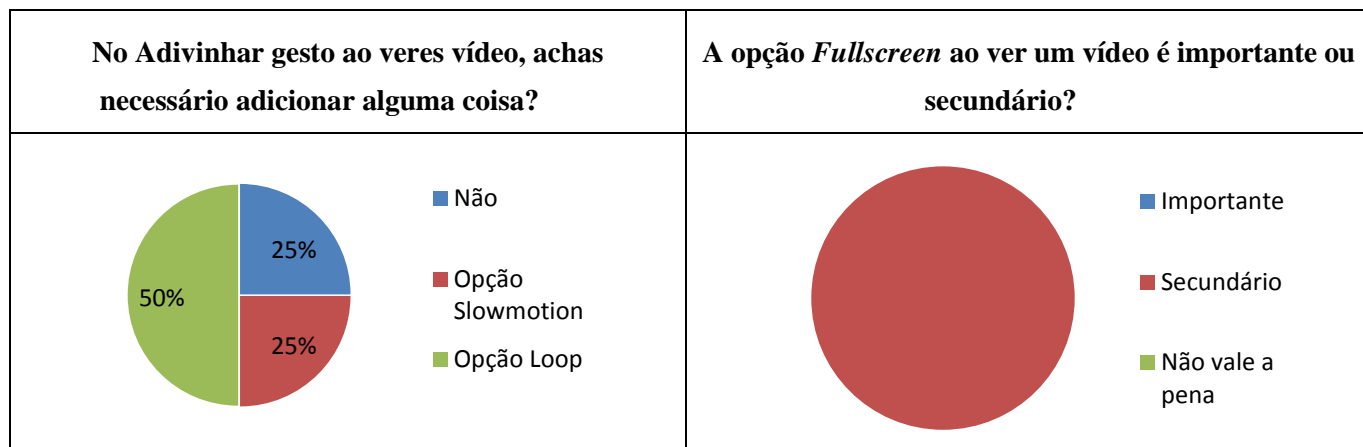


Figura 89 – Gráficos em círculo de respostas do segundo questionário

Pelos dados dos gráficos da *Figura 89*, revela-se a necessidade de melhorar com mais funcionalidades o controlo de vídeo no Adivinhar gesto. Em termos de *fullscreen*, a pergunta surgiu para compreender as necessidades do público-alvo e saber era necessário implementar esta funcionalidade, só existente no Adivinhar gesto.

6.5.3 Terceiro questionário – Perguntas genéricas

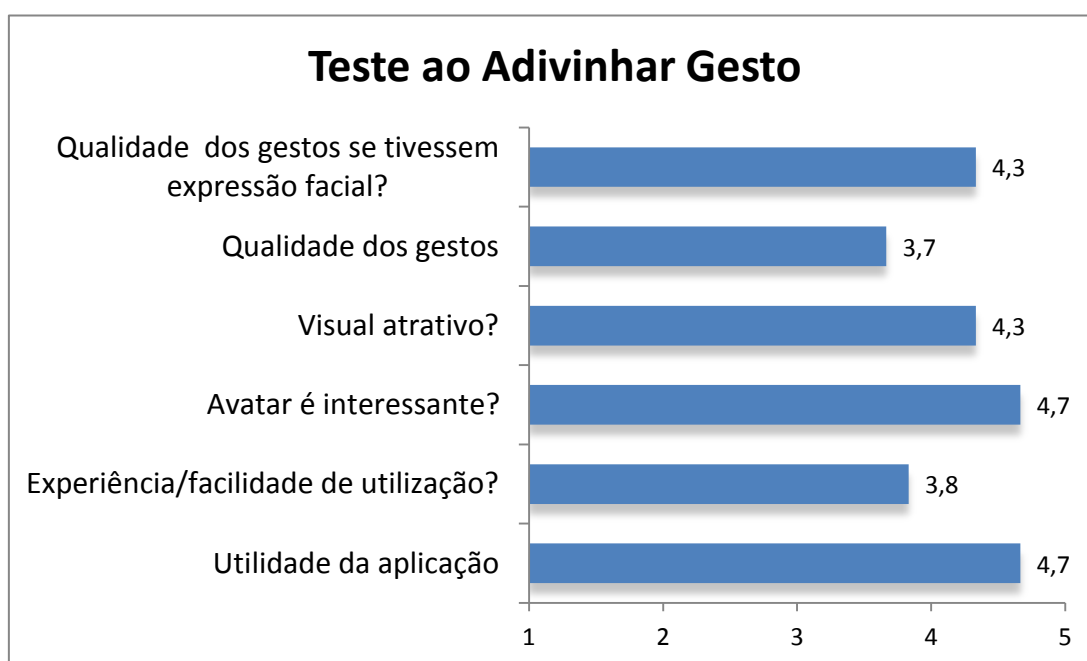


Figura 90 – Gráfico da média dos resultados para o Questionário Geral

Na *Figura 90*, a resposta à pergunta “*Qualidade dos gestos se tivessem expressão facial?*” permite saber até que ponto é importante a expressão facial, relativamente à qualidade dos gestos já implementados que não tem essa característica. O conjunto das respostas deste questionário permite-me resumir a aplicação desta forma:

Pontos fortes

- Potencial e utilidade da aplicação para introduzir conceitos básicos de LGP
- Avatar chama a atenção e prende o interesse
- Capacidade de complementar o vídeo por ser capaz de utilizar gestos atómicos em frases

Pontos fracos

- Qualidade dos gestos e facilidade de utilização

Relativamente aos pontos fracos, os gestos devem ser validados por um surdo experiente antes da divulgação ao público. A facilidade de utilização da aplicação é apenas boa, derivado dos problemas identificados no modo Adivinhar. Depois de corrigidos esses problemas, penso que a experiência de utilização ficará muito boa.

A pergunta sobre “*Qualidade dos gestos se tivessem expressão facial?*” tem como objetivo estimar qual a diferença que faz para um surdo ver o avatar com e sem expressão facial. Desta forma é possível traçar se é um objetivo prioritário, a adição de expressão facial. Depois de falar com um participante, ele disse que as expressões faciais aplicadas às palavras são indiferentes, porém, nas frases a expressão facial já faz toda a diferença tanto no sentido da frase, como no nível de satisfação retirado pela animação.

O aspeto do *interface* da aplicação foi considerado bom, quase muito bom, logo, não é uma prioridade a curto-prazo melhorar essa vertente.

6.6 Sumário

Este capítulo apresentou os resultados da avaliação por utilizadores. Os resultados evidenciaram os pontos fracos e fortes da aplicação “*Os meus primeiros gestos*”.

Capítulo 7

Conclusão

Os **objetivos** traçados para o trabalho descrito nesta tese foram:

1. Conceber um avatar animado capaz de produzir gestos de Língua Gestual Portuguesa, cujo processo de animação é assistido pela captura de movimentos
2. Combinar gestos atómicos de modo a produzir frases simples em LGP;
3. Desenvolver uma aplicação que ilustre as vantagens do uso de avatares como intérpretes virtuais de LGP.

O primeiro objetivo desta tese consistia na geração de um avatar e a manipulação do seu esqueleto com o intuito de produzir animações de gestos de Língua Gestual Portuguesa. No domínio da animação, esta pode ser feita, pelo menos de duas formas: manipulação manual recorrendo a um *software* de animação; ou recorrendo a sensores de movimentos para captura de dados de movimento e posterior integração destes no modelo virtual. A última forma resulta em animações com movimentos mais credíveis e realistas que a anterior.

O objetivo ideal seria recorrer totalmente à técnica de captura de movimentos para reproduzir um gesto nas três componentes essenciais da interpretação de uma língua gestual, nomeadamente, corpo, mãos e face. O sensor de movimentos *Microsoft Kinect*, uma tecnologia recente e em evolução, permitiu implementar o processo de captura de movimentos de um humano, e a conversão desse movimento para um avatar. Todavia, este sensor não tem suporte nativo à captura da localização dos dedos, o que implicou a investigação de um meio para lidar com esse desafio.

A conceção de um algoritmo capaz de detetar as articulações dos dedos em tempo real, suficientemente robusto para ter em conta as oclusões existentes nas várias configurações possíveis da mão, não era um desafio trivial, sendo objeto de estudo de outras teses (Oikonomidis, Kyriazis e Argyros 2011). Este processo não sendo um objetivo essencial, levou a investigação noutra direção. Procurou-se um método semiautomático que envolvesse a participação do utilizador para detetar os dedos, marcando numa foto a posição das articulações de cada dedo.

Posteriormente à marcação dos dedos na foto, as coordenadas de cada articulação seriam convertidas do espaço 2D para o espaço 3D.

Esta investigação resultou num *software* capaz de converter, com o auxílio do utilizador, a pose estática 3D do corpo e mãos para um esqueleto em *MotionBuilder*. Contudo, a investigação não avançou da captura de gestos de letras para gestos de palavras devido, sobretudo, ao facto deste método não ser prático para marcar a posição dos dedos das duas mãos em vários instantes do tempo. Assim sendo, o percurso desta tese tomou outra direção.

A abordagem adotada consistiu em animar o avatar por um método híbrido, utilizando dois tipos de manipulação do avatar. Este método consiste em, por um lado capturar o movimento do tronco com o *Kinect*, e por outro lado, animar de forma manual por *keyframe* o movimento da mão por cima da animação capturada com o *Kinect*. Foi identificado durante este processo um *pipeline* para a criação dos gestos no avatar: captura do movimento do corpo, ligação do avatar aos dados do Kinect, tratamento de dados, e adição da animação manual do movimento da mão e dedos.

O resultado desta primeira fase consistiu numa animação fluída de alguns gestos para serem avaliados por indivíduos proficientes em Língua Gestual Portuguesa. Por conseguinte, foi concebido e submetido a avaliação um primeiro protótipo composto por uma aplicação capaz de exibir a tradução de qualquer letra para LGP, e a visualização de um conjunto de frases em LGP feitas a partir da combinação de gestos atómicos. Este protótipo concretizou o primeiro e o segundo objetivo desta tese, a animação de um avatar e a composição de frases respetivamente.

A avaliação inicial revelou grandes potencialidades na tecnologia por parte dos surdos, tendo estes sugerido a aplicação ao ensino de crianças, no entanto, apontaram algumas limitações, por exemplo, no avatar. Nesse sentido, procurei um avatar com características mais adequadas, entre as quais, mais contraste entre mão e braço, roupas lisas e sem adereços. Daí ter surgido o avatar *Nucha*, que foi avaliado e considerado adequado à interpretação de LGP. Posteriormente desenvolveu-se uma aplicação de modo a expor as funcionalidades do uso de um avatar 3D aplicado à Língua Gestual, animado com movimentos humanos.

Seguindo a sugestão dada pela comunidade surda, optei por desenvolver uma aplicação *Web* que introduz conceitos básicos em LGP de uma forma lúdica, dirigida a crianças. Após investigar o método de ensino das crianças surdas, concluí que o método predominante é o ensino bilingue (Língua Portuguesa e LGP). Verifiquei também que existe mais dificuldade na aprendizagem de português escrito, por parte de crianças surdas do que por ouvintes, porque a sua língua materna é diferente do Português (Baptista 2010). Portanto, perante esta dificuldade cheguei a um conceito fundamental: a aprendizagem de crianças surdas e para surdos no geral, deve necessariamente contemplar a associação entre a palavra, o gesto, e uma imagem representativa do conceito.

O resultado final foi a concretização do terceiro e último objetivo desta tese. A criação de uma prova de conceito através da demonstração de um dicionário de gestos *online*, que introduz

conceitos básicos de LGP, como palavras, frases simples, a sua construção frásica e uma componente didática para teste de conhecimentos. O público-alvo é abrangente na medida em que pode ajudar qualquer indivíduo que queira obter conhecimentos básicos de LGP, designadamente crianças ou adultos. Porém, também pretende ajudar na associação de conceitos a palavras da Língua Portuguesa, destinado sobretudo a crianças surdas. A aplicação não pretende ensinar LGP, nem substituir nenhum professor no seu ensino, mas complementar a aprendizagem através de uma ferramenta divertida, e em simultâneo constituir um dicionário de gestos *online*.

A avaliação desta aplicação revelou resultados muito interessantes, e conclui que é uma aplicação com muitas potencialidades.

Como etapa final, investigou-se como capturar e integrar expressões faciais num avatar e integrou-se em algumas palavras, designadamente nos pronomes interrogativos, as expressões de surpresa.

Durante o desenvolvimento da tese, concluí que as grandes vantagens do uso de avatares em detrimento do vídeo são:

- O uso de um elemento original e cativante, um boneco animado como intérprete de LGP
- A vantagem da combinação (exponencial) de gestos atómicos de modo a formar frases

A grande desvantagem é o tempo de produção de um gesto. Enquanto em vídeo se demora apenas o tempo de execução do próprio gesto, em animação, embora esteja dependente da complexidade do gesto e da experiência do animador, verifiquei um tempo de produção de meia hora para a animação de cada gesto.

7.1 Dificuldades encontradas

A maior dificuldade foi encontrar o caminho que levou à solução final do projeto. A natureza da investigação sugeriu-me vários caminhos à medida que fui avançando neste trabalho. A minha experiência anterior incidiu em projetos práticos com objetivos predefinidos e fixos. Por essa razão foi difícil passar para uma metodologia de investigação, que assume a redefinição de objetivos e metas conforme os resultados que vão sendo obtidos. *Avatares em LGP* é um tema vasto com uma grande diversidade de aplicações, e a abrangência deste domínio constituiu para mim o maior desafio.

A animação 3D é uma área menos focada na Licenciatura e Mestrado em Engenharia Informática. Seria interessante ter usufruído de mais disciplinas na área de modelação e animação 3D.

O contacto com tecnologias recentes, como o *Kinect* e o *Leap Motion*, deu-me a oportunidade de investigar áreas pouco exploradas e em perceber até onde é possível chegar com este *hardware*.

O mundo surdo é muito diferente do mundo dos ouvintes. A percepção do mundo por parte dos surdos implica maneiras diferentes de pensar, e por conseguinte, têm necessidades diferentes. O que achamos certo para os surdos pode ser completamente diferente. Por outro lado, a comunicação com surdos tem de ser acompanhada por intérpretes (salvo algumas exceções), o que gera barreiras na comunicação e cria obstáculos na compreensão do público-alvo. A burocracia envolvente ao contato com surdos pode ser complicada. Quando contactei com a APS, o processo tornou-se demorado, foi necessário delegar um intérprete e depois agendar conforme a disponibilidade de todos. Observei que por vezes, foi um pouco difícil chegar à comunidade, pois realizei algumas tentativas de contato com surdos que não foram bem-sucedidas.

7.2 Opinião Crítica

Considero que ainda há muito trabalho para fazer no desenvolvimento da aplicação. Foi desenvolvida uma prova de conceito que demonstra que o sistema tem utilidade, mas é necessário carregar várias palavras, vários temas, para de facto se tornar num produto fora do “laboratório”. Foi demonstrado pelos resultados da avaliação que a funcionalidade de compor frases com gestos é muito interessante, portanto, esse deve ser um dos caminhos prioritários. No entanto, a aplicação pode evoluir muito mais, com adição de novas funcionalidades e jogos. Um dos públicos-alvo de “*Os meus primeiros gestos*” são as crianças, logo, a aplicação deve ser testada por elas.

O Microsoft Kinect[®] é um sensor focado para o *tracking* do corpo, e não para as mãos ou dedos, isto implica, a animação por um método híbrido, sendo o ideal a utilização de um método que capture além do corpo, também a posição dos dedos, por exemplo, com luvas. Por vezes, o *tracking* tem falhas e isto obriga que seja feito um trabalho intenso de tratamento de dados para tornar a captura numa animação usável num avatar. A tecnologia utilizada pode evoluir para sistemas com melhor desempenho de *tracking*, e assim, permitir a captura da posição dos dedos e tornar as animações ainda mais credíveis e com um tempo de produção inferior.

Algo a explorar melhor, será a componente do rosto. Na LGP esta componente é importante particularmente nas frases, logo, para ter uma aplicação com mais qualidade na interpretação de LGP, é preciso estender esta animação a mais palavras e incluí-la em mais frases com expressões bem destacadas.

7.3 Trabalho futuro

- Resolver os problemas identificados na última avaliação por utilizadores.
- Permitir a criação dinâmica de frases sem que estas tenham de existir já predefinidas no sistema.

- Otimizar o processo de captura de gestos, por exemplo: estudar a opção *iPi Soft* que usa dois *Kinects* e depois importar a animação para o *MotionBuilder*. Com duas câmeras os dados poderão não ter que ser sujeitos a tanto tratamento, logo, menos tempo no *pipeline* da produção da animação e ao mesmo tempo animações mais fluidas.
- Otimizar a aplicação com a criação de um sistema de *cache*, no *browser* ou no servidor. De modo a prepara-la para grande quantidade de dados.
- Carregar a aplicação com muitos dados. Adicionar vários gestos, palavras, gestos, e temas.
- Garantir a operabilidade da aplicação em vários *browsers*.
- Testar o sistema com crianças surdas.
- Estar atento a novas tecnologias, e aguardar a chegada da segunda versão do Kinect esperada para 2014, mais poderosa que o primeiro.
- Utilizar um avatar diferente dos que foi desenvolvido, e fazer a migração da animação para esse, de modo a dar a escolha ao utilizador do avatar que pretende ver.
- Criar novos jogos, diferentes do Adivinhar gesto.
- Reconhecer o gesto, tutorial. Gesto correto e gesto incorreto. Traduzir texto em LGP. Filmes de animação com avatares mesmo surdos, combinando as frases.
- Estudar a entoação nos gestos, e introduzir uma componente ao gesto para dar personalidade a cada avatar em tempo real, alterando pequenos movimentos ou adicionando pequenas animações aleatórias. Um exemplo seria aumentar ou diminuir a velocidades dos gestos para dar a entender que o avatar revela personalidade, cansaço ou outros fatores interessantes
- Fazem falta no mundo surdo aplicações *Web* dirigidas a eles. Não só dicionários, ou aplicações de suporte ao ensino, mas aplicações que façam a ponte entre serviços e surdos. Por exemplo, aplicações que traduzam texto para LGP, ou aplicações que traduzam voz para LGP: A tecnologia de animação do avatar desenvolvida até aqui, pode partir como base para esses projetos.

- Desenvolver aplicação para Android ou iPad de “*Os meus primeiros gestos*”
- Usar o Unity 3D para introduzir o avatar no *browser* dentro de um plugin em vez de ser num vídeo. Isto permitia interagir mais com o *browser* como rodar a câmara, fazer *zoom in* ou *out*, e outras funcionalidades.

Bibliografia

- Associação de Surdos da Alta Estremadura.* s.d. http://associacaosurdosaltaestremadura.org/index.php?id=44&_lang=PT (acedido em 10 de Janeiro de 2013).
- Autodesk. *Maya*. 2011. <http://www.autodesk.com/products/autodesk-maya> (acedido em 22 de Setembro de 2012).
- . *MotionBuilder Software*. 2011. <http://www.autodesk.com/products/motionbuilder/> (acedido em 22 de Outubro de 2013).
- . *Project Pinocchio*. 2013. <http://projectpinocchio.autodesk.com/> (acedido em 20 de Junho de 2013).
- Baltazar, Ana Bela. *Dicionário de Língua Gestual Portuguesa*. Porto: Porto Editora, 2010.
- Baptista, Maria Madalena Belo da Silveira. *Alunos Surdos: Aquisição da Língua Gestual e Ensino*. 9 de Março de 2010.
- Blender Foundation. *Blender*. 2011. <http://www.blender.org/> (acedido em 22 de Setembro de 2012).
- Brekelmans, Jasper. *Brekel Kinect*. 2011. <http://www.brekel.com/kinect-3d-scanner/> (acedido em 5 de Novembro de 2012).
- Camargo, Alessandra. *Libras na Educação Infantil*. 3 de Julho de 2011. <http://librasnaeducacaoinfantil.blogspot.pt/2011/07/dicas-para-trabalhar-com-auluno-surdo.html> (acedido em 2013).
- da Silva, Renata Cristina. “Apropriações do termo avatar pela Cibercultura: do contexto religioso aos jogos eletrônicos.” *Contemporânea* 8. 2010. 120-131.
- Damer, Bruce, Jeremy Judson, e Jackie Dove. *Avatars!; Exploring and Building Virtual Worlds on the Internet*. Peachpit Press, 1999.

- DAZ 3D Inc. *DAZ 3D*. 2011. <http://www.daz3d.com/> (acedido em Outubro de 30 de 2013).
- Deusdado, Leonel Domingues. “Ensino da Língua Gestual Assistido por Personagens 3D Virtuais.” *Biblioteca Digital IPB*. 2002. https://bibliotecadigital.ipb.pt/bitstream/10198/1211/1/TeseMestrado_Leonel.pdf (acedido em 5 de Março de 2013).
- Dinis, José Maria Cardoso Carneiro. “Noção de captura de movimento e a sua utilização no meio artístico.” 2012.
- DoIT. *Software Development Methodology of New Hampshire*. 2005. <http://www.nh.gov/doit/internet/standards/documents/AppendixE-SoftwarDevelopmentMethodologyReference021805.pdf> (acedido em 22 de Outubro de 2013).
- Easy Mocap. *Easy Mocap*. 2011. <http://mocap.cguse.com/en/> (acedido em 5 de Novembro de 2012).
- Edward, Klima, e Bellugi Ursula. “The Signs of Language.” *Harvard University Press*, 1979.
- Farzad, Pezeshkpour, Marshall Ian, Elliott Ralph, e Bangham J. Andrew. “Development of a Legible Deaf-Signing Virtual Human.” *Multimedia Computing and Systems IEEE International Conference*. 1999. 333-338.
- Hash, Inc. *Hash Animation:Master*. 2011. <http://www.hash.com/> (acedido em 22 de Setembro de 2012).
- Higuchi, Yu. *Vocaloid Promotion Video Project*. 2008. http://www.geocities.jp/higuchuu4/index_e.htm (acedido em 5 de Novembro de 2012).
- iPi Soft. *iPi Soft*. 2011. <http://ipisoft.com/> (acedido em 5 de Novembro de 2012).
- Kennaway, Richard. “Synthetic animation of deaf signing gestures.” *Gesture and Sign Language in Human-Computer Interaction*. Springer Berlin Heidelberg, 2002. 146-157.
- Kitagawa, Midori, e Brian Windsor. *MoCap for artists: workflow and techniques for motion capture*. Taylor & Francis US, 2008.
- Lee, Seungyon, Valerie Henderson, e Helene Brashear. “CopyCat : Learning American Sign Language (ASL) through a Gesture-Based Computer Game.” *Student Design Competition International Conference*. 2005.

- Leonor. *LGP, Comunidade e Cultura Surda*. 29 de Abril de 2013. <http://lgpcomunidadeeculturasurda.blogspot.pt/2013/04/origem-da-lingua-gestual-portuguesa.html> (acedido em 26 de Outubro de 2013).
- Makehuman TM. *Makehuman*. 2011. <http://www.makehuman.org/> (acedido em 30 de Outubro de 2013).
- Martins, Mariana, José Pedro Ferreira, e Ana Mineiro. *Os Dicionários e os Avatares Gestuais*. Lisboa: Universidade Católica Editora, 2012.
- Measurand Inc. *Motion Capture Resources - History*. 2011. <http://www.motion-capture-system.com/resources/history.html> (acedido em 14 de Outubro de 2013).
- Naugle, Lisa Marie. "Motion Capture: Re-collecting the Dance." *Proceedings of the Twenty-First Biennial Conference*. Barcelona, Spain, 1999. 208-213.
- NI MATE. *NI mate and Blender*. 2011. www.ni-mate.com/use/blender/ (acedido em 5 de Novembro de 2012).
- Oikonomidis, Iason, Nikolaos Kyriazis, e Antonis A. Argyros. "Efficient model-based 3D tracking of hand articulations using Kinect." *BMVC*. 2011. 1-11.
- Power, Des, e Mary Power. "SIGN LANGUAGE AVATARS A NEW METHOD FOR DEAF PEOPLE TO ACCESS MEDIA COMMUNICATION." *MEDIA ACCESS REVIEW*, s.d.
- Programa Leonardo da Vinci. *Spread the Sign*. 2008. <http://www.spreadthesign.com/> (acedido em 5 de Fevereiro de 2013).
- Ralph, Elliott, Glauert John RW, Kennaway J.R., e Marshall Ian. "The Development of Language Processing Support for the ViSiCAST Project." *Proceedings of the fourth international ACM conference on Assistive technologies*. 2000. 101-108.
- Reallusion. *iCone 5*. 2011. <http://www.reallusion.com/iclone/> (acedido em 5 de Novembro de 2012).
- Signing Savvy LLC. *Signing Savvy*. 2009. <http://www.signingsavvy.com> (acedido em 23 de Junho de 2013).
- Smith Micro Software, Inc. *Poser 3D*. 2011. <http://poser.smithmicro.com/> (acedido em 22 de Setembro de 2012).

- Stegmueller, Stefan. *Candescent NUI*. 18 de Março de 2012.
<http://candescentnui.codeplex.com/> (acedido em 20 de Janeiro de 2013).
- Stephen Cox, Michael, et al. "TESSA, a system to aid communication with deaf people."
Proceedings of the fifth international ACM conference on Assistive technologies.
 2002. 205-212.
- ValorVisual. *Curso de Língua Gestual Portuguesa em DVD*. 2009.
<http://www.languagegestual.com/index.php> (acedido em 5 de Março de 2013).
- VCom3D. *Signing Math & Science Dictionaries*. s.d. <http://signsci.test.terc.edu> (acedido em 10 de Junho de 2013).
- Verlinden, Margriet, Corrie Tijsseling, e Han Frowein. "A Signing Avatar on the WWW."
Gesture and Sign Language in Human-Computer Interaction. 2002. 169-172.
- Vygotsy, Lev S. *Pensamento e linguagem*. 1989.
- Wang, Robert, Chris Twigg, e Kenrick Kin. *Gestural User Interfaces - 3Gear Systems*. 2012.
<http://www.threegear.com/> (acedido em 26 de Outubro de 2013).
- Wikipedia. *Língua gestual portuguesa*. 2007.
http://pt.wikipedia.org/wiki/Língua_gestual_portuguesa (acedido em 30 de Junho de 2013).
- Zafrulla, Zahoor, Harley Hamilton, Helene Brashear, Peter Presti, e Thad Starner. "American Sign Language Recognition with the Kinect." *Proceedings of the 13th international ACM conference on multimodal interfaces*. 2011. 279-286.
- Zwitserslood, Inge, Margriet Verlinden, Johan Ros, e Sanny van der Schoot. "Synthetic signing for the deaf: eSign." 2005.

ANEXOS

Anexo 1 – Contexto sociocultural dos surdos em Portugal

Em Portugal existem 33 mil surdos (Associação de Surdos da Alta Estremadura s.d.). Desde o seu aparecimento, no ano de 1823 (Leonor 2013), a língua gestual era proibida. Os surdos usavam-na clandestinamente para comunicar. Nas escolas de ensino regular não havia intérpretes e só em 1997 foi reconhecida como língua oficial de Portugal a Língua Gestual Portuguesa.

O pensamento está ligado intimamente à linguagem (Vygotsy 1989). A forma de pensar de um ouvinte diverge da forma de pensar de um surdo. Ao passo que um surdo pensa “visualmente”, quem se expressa numa língua oral tem um pensamento mais abstrato. Logo, estas diferenças marcam a dificuldade dos surdos na adaptação ao universo das línguas escritas.

Ainda existe uma grande quantidade de surdos analfabetos, muitos deles não sabem sequer língua gestual. Falam através de mímica e comunicam assim no seu dia-a-dia. Esta situação é mais predominante nas gerações anteriores. Porém, hoje em dia existe uma grande diferença para melhor relativamente à institucionalização e educação dos surdos.

A língua gestual assume, portanto, uma importância fundamental para o surdo, para o seu ambiente familiar, profissional e cultural. E envolve outras pessoas que com ele têm de comunicar. Resumindo, a língua gestual faz parte da cultura do surdo, tal como para um cidadão português, por exemplo, a língua portuguesa faz parte da sua cultura.

Anexo 2 – Especificações técnicas e algoritmo usado pelo Kinect

A câmera de cor RGB permite por defeito resoluções de 640x480 a 30 *frames* por segundo, porém é possível ir até 1280x1024 a 12 *frames* por segundo. A resolução de profundidade suporta resoluções entre 80x60, 320x240 e 640x480 a 30 *frames* por segundo. O reconhecimento do sensor de profundidade tem um alcance de 0.8 a 4 metros para a versão Xbox enquanto a versão especializada para PC, tem um alcance de 0.5 a 3 metros.

A capacidade de fazer o reconhecimento automático das articulações, e por conseguinte, construir um esqueleto virtual do utilizador, advém da captação dos dados de profundidade.

O *hardware* por detrás do Kinect foi inicialmente desenvolvida pela empresa PrimeSense, do qual a Microsoft adquiriu posteriormente essa tecnologia. O processo que a câmera utiliza para obter a profundidade chama-se *structured light*, e consiste na projeção de um padrão de pontos infravermelhos no espaço. A forma como esses pontos se deformam em contacto com a superfície, permite calcular a distância do emissor (sensor) ao objeto.

Anexo 3 – Questionário da avaliação do protótipo inicial para componente do Soletrar palavras em LGP



Fase 1 - Soletrar:

Palavras	Que palavra reconheceu?	Qualidade	Observações
Palavra 1		1 <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> 3	
Palavra 2		1 <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> 3	
Palavra 3		1 <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> 3	
Palavra 4		1 <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> 3	
Palavra 5		1 <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> 3	

1 – Tem utilidade ou aplicação o soletrar palavras? (Nível de utilidade de 1 a 5)

2 – Onde imagina ver aplicado um exemplo deste tipo? (ensino a crianças? Por exemplo como?)

3 – O que seria bom melhorar? (animação, boneco em si, expressividade do boneco, realismo, cenário de fundo mais adequado)

Anexo 4 – Questionário da avaliação do protótipo inicial para a visualização de frases em LGP



Fase 2 - Frases:

Frases	Que frase reconheceu?	Qualidade	Observações
Frase 1		1 <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> 3	
Frase 2		1 <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> 3	
Frase 3		1 <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> 3	
Frase 4		1 <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> 3	
Frase 5		1 <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> 3	
Frase 6		1 <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> 3	

Fase 2 - Frases

1 – Tem utilidade ou aplicação o soletrar palavras? (Nível de utilidade de 1 a 5)

2 – Onde imagina ver aplicado um exemplo deste tipo? (ensino a crianças? Por exemplo como?)

3 – O que seria bom melhorar? (animação, boneco em si, expressividade do boneco, realismo, cenário de fundo mais adequado)

4 – De 1 a 3 como classificaria o nível de realismo do exemplo apresentado? E qual o nível para o exemplo anterior, o soletrar palavras?

5 – Comparando o exemplo anterior com este, o que achou da mudança do boneco? Qual o melhor?

6 – No exemplo anterior, o boneco tinha muito pouca expressividade ou nenhuma em relação a este exemplo. Concorde? Qual a importância da expressividade na passagem da mensagem através do boneco? O que acha que poderia melhorar neste aspecto?

Anexo 5 – Descrição formal do caso de uso Traduzir e Adivinhar gesto de palavras

Caso de Uso	Traduzir palavra em Língua Gestual
Sumário	O utilizador visualiza animação de um avatar a comunicar língua gestual de uma determinada palavra
Actor	Criança
Pré-condição	Utilizador seleccionou modo consulta
Descrição	<ol style="list-style-type: none"> 1. Utilizador escolhe o tema que quer consultar. 2. Sistema carrega e mostra todas as palavras que tem para o tema escolhido. 3. Utilizador escolhe a palavra que quer consultar.
Alternativas	<p>2.a - Sistema não tem palavras para tema</p> <p>2.b - Sistema avisa utilizador que não tem palavras para ele tema e pára</p>
Pós-condição	Utilizador vê animação do gesto correspondente à palavra escolheu

Tabela 8 – Descrição formal do caso de uso Traduzir palavra em LG

Caso de Uso	Adivinhar gesto para uma determinada palavra
Sumário	O utilizador tenta adivinhar de entre várias animações de gestos de LGP qual o gesto que corresponde a uma determinada palavra
Actor	Criança
Pré-condição	Utilizador seleccionou modo adivinha
Descrição	<ol style="list-style-type: none"> 1. Utilizador escolhe o tema para a adivinha. 2. Sistema carrega e mostra todas as palavras que tem para o tema escolhido. 3. Utilizador escolhe uma palavra. 4. Sistema carrega aleatoriamente animações de gestos além daquela que corresponde à palavra escolhida. 5. Utilizador visualiza todos os gestos. 6. Utilizador selecciona o gesto que corresponde à palavra.

Alternativas	<p>2.a - Sistema não tem palavras para tema</p> <p>2.b - Sistema avisa utilizador que não tem palavras para aquele tema e pára</p> <p>6.a - Gesto selecionado não corresponde à palavra</p> <p>6.b - Sistema avisa utilizador que está errado e solicita nova tentativa</p>
Pós-condição	Sistema avisa que utilizador acertou na escolha e volta à lista de palavras

Tabela 9 – Descrição formal do caso de uso Adivinhar gesto para uma determinada palavra

Anexo 6 – Descrição formal do caso de uso Compositor de Frases

Caso de Uso	Traduzir frase em Língua Gestual
Sumário	O utilizador visualiza animação de um avatar a comunicar língua gestual de duas palavras de temas diferentes
Actor	Criança
Pré-condição	Utilizador seleccionou modo consulta
Descrição	<ol style="list-style-type: none"> 1. Utilizador escolhe o tema que quer consultar. 2. Sistema carrega e mostra todas as palavras que tem para o tema escolhido. 3. Utilizador escolhe a primeira palavra que quer consultar. 4. Sistema carrega animação com a palavra escolhida. 5. Utilizador escolhe outro tema. 6. Sistema carrega e mostra todas as palavras para o tema escolhido. 7. Utilizador clica com o segundo botão do rato ou fica com o rato algum tempo em cima da palavra pretendida. 8. Sistema mostra um botão para permitir combinar a palavra seleccionada anteriormente com a actual. 9. Utilizador clica nesse botão.
Alternativas	<p>2.a - Sistema não tem palavras para tema</p> <p>2.b - Sistema avisa utilizador que não tem palavras para ele tema e pára</p> <p>6.a - Sistema não tem palavras para tema</p> <p>6.b - Sistema avisa utilizador que não tem palavras para ele tema e pára</p> <p>9.a - Utilizador tenta combinar duas palavras inválidas</p> <p>9.b - Sistema avisa o utilizador do erro e pára</p>
Pós-condição	Utilizador vê animação do gesto correspondente à combinação das duas palavras que escolheu

Tabela 10 – Descrição formal do caso de uso Traduzir frase em LG

Anexo 7 – Descrição dos *PageMethods* utilizados na aplicação *front-end*

`string UpdateNavTabs(string id)`

`string UpdateLeftBar(string text, string themeId, string videoType)`

`string UpdateContentBar(string text, string themeId, string videoType)`

`SentenceQueryResult IsValidWord(string[] words)`

`string GetRandomSentences(int rows)`

`GestureVideo[] GetRandomGestures(string differentFrom, string differentFromVideo, string neId, string videoType, string rows)`

UpdateNavTabs

Este método vai buscar conteúdo da *div* dos temas, do qual `string id` corresponde *id* do tema que o utilizador seleccionou.

UpdateLeftBar

Este método vai buscar o conteúdo para a *div* da secção da lista de palavras, do qual `string text` corresponde à *string* de procura que utilizador escreveu, `string themeId` corresponde ao *id* do tema activo, e `string videoType` corresponde ao tipo de vídeo que o utilizador pretende visualizar.

UpdateContentBar

Este método vai buscar o conteúdo da *div* principal, onde se selecciona as palavras, do qual `string text` corresponde à *string* de procura que utilizador escreveu, `string themeId` corresponde ao *id* do tema activo, e `string videoType` corresponde ao tipo de vídeo que o utilizador pretende visualizar.

isValidWord

Este método verifica se a frase que o utilizador está a tentar combinar corresponde a alguma combinação de palavras existentes, do qual `string[] words` corresponde a um *array* com as palavras que o utilizador sequencialmente combinou no compositor.

Se não corresponder, retorna *null*, se corresponder mas não combinar com a frase completa, retorna uma instância de `SentenceQuery`.

Se corresponder a uma frase completa, então retorna uma instância de `SentenceQuery` com os atributos correctamente preenchidos.

GetRandomWords

Este método retorna uma lista de frases aleatórias das frases existentes no sistema, do qual `int rows` corresponde ao número de frases que se pretende receber.

GetRandomGestures

Este método retorna um *array* de objetos `GestureVideo` contendo vídeos de gestos aleatórios, do qual `string differentFrom` corresponde à palavra que utilizador quer adivinhar, `string differentFromVideo` ao vídeo da palavra que utilizador quer adivinhar, `string themeId` ao *id* do tema que a palavra corresponde, `string videoType` ao tipo de vídeo que o utilizador pretender ver e `string rows` o número de vídeos com que vamos misturar/baralhar o vídeo certo.

Anexo 8 – Resultados da aplicação (algumas *screenshots*)

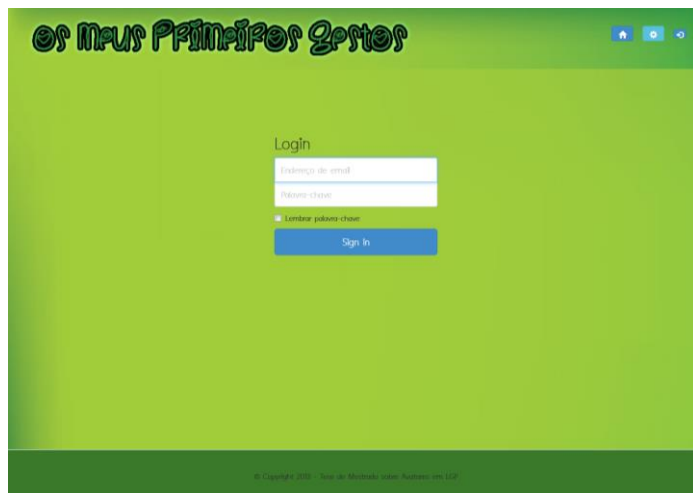


Figura 91 – Interface Login (front-end)



Figura 92 – Interface Home (front-end)

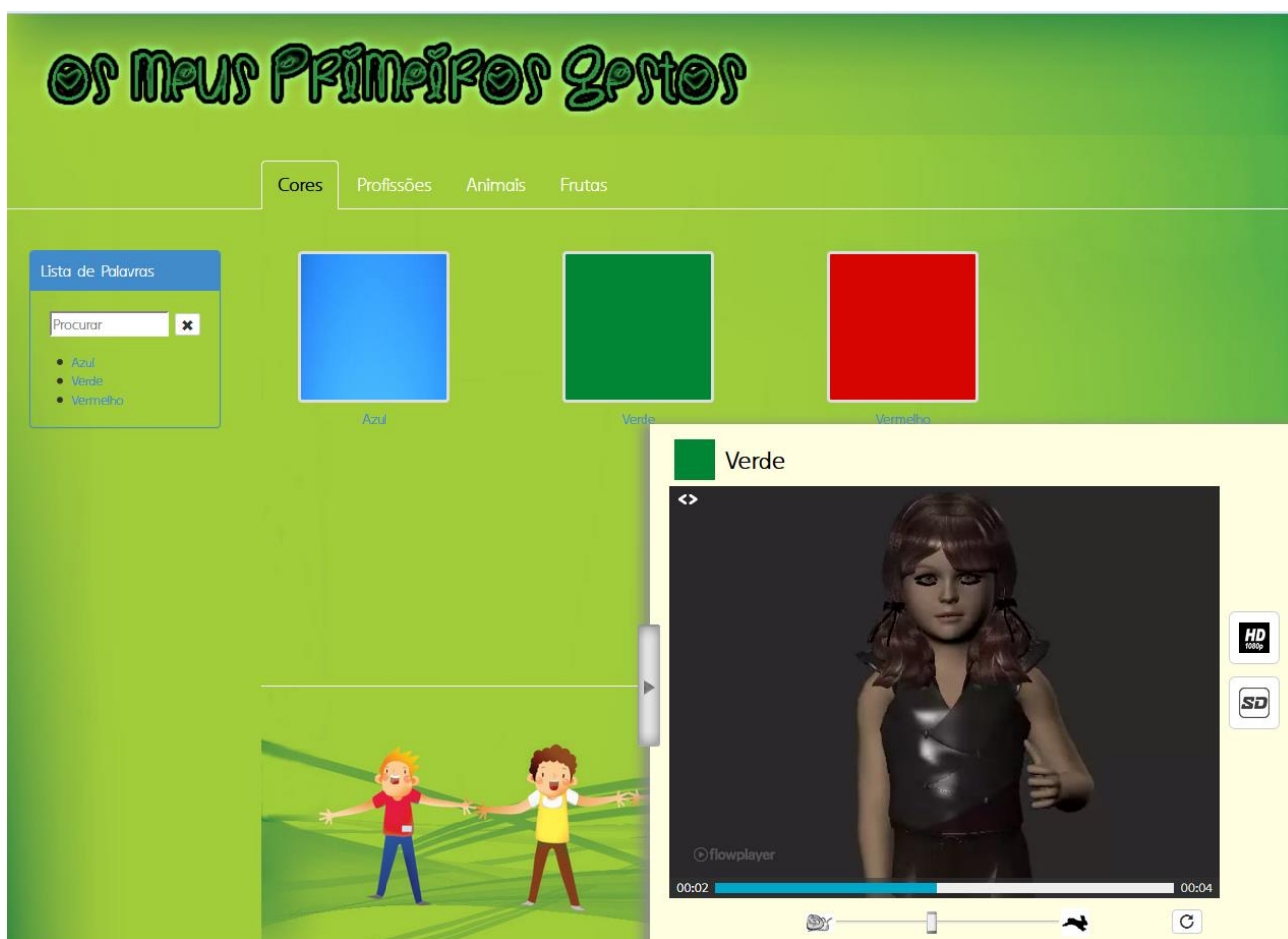


Figura 93 - Interface Dicionário, ver gesto (front-end)

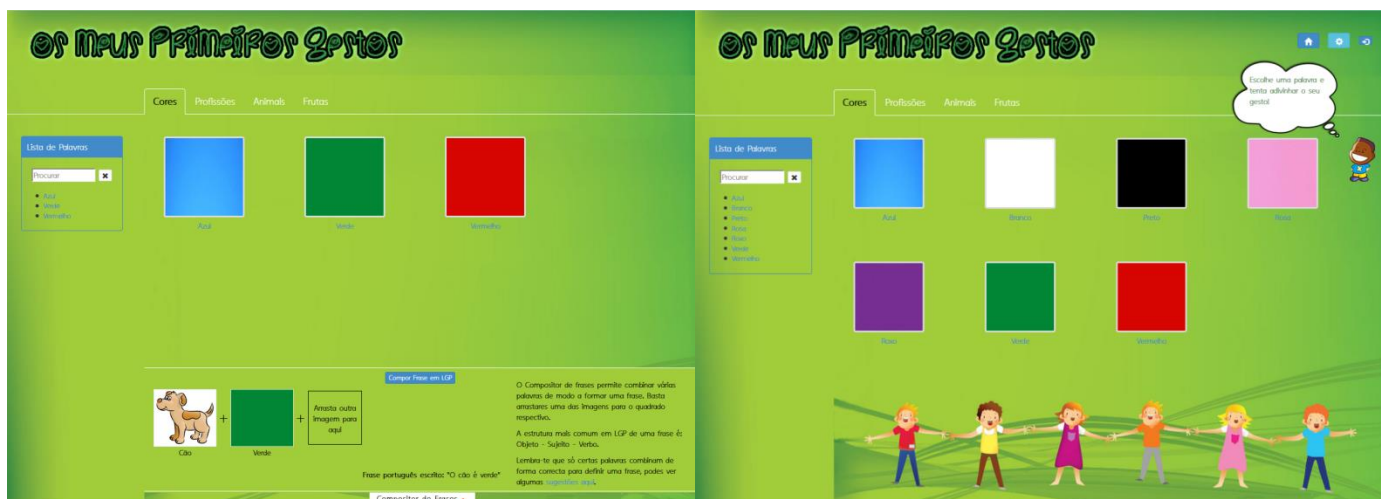


Figura 94 – Interface Dicionário:
compositor de frases (front-end)

Figura 95 - Interface Adivinhar Gesto:
início (front-end)

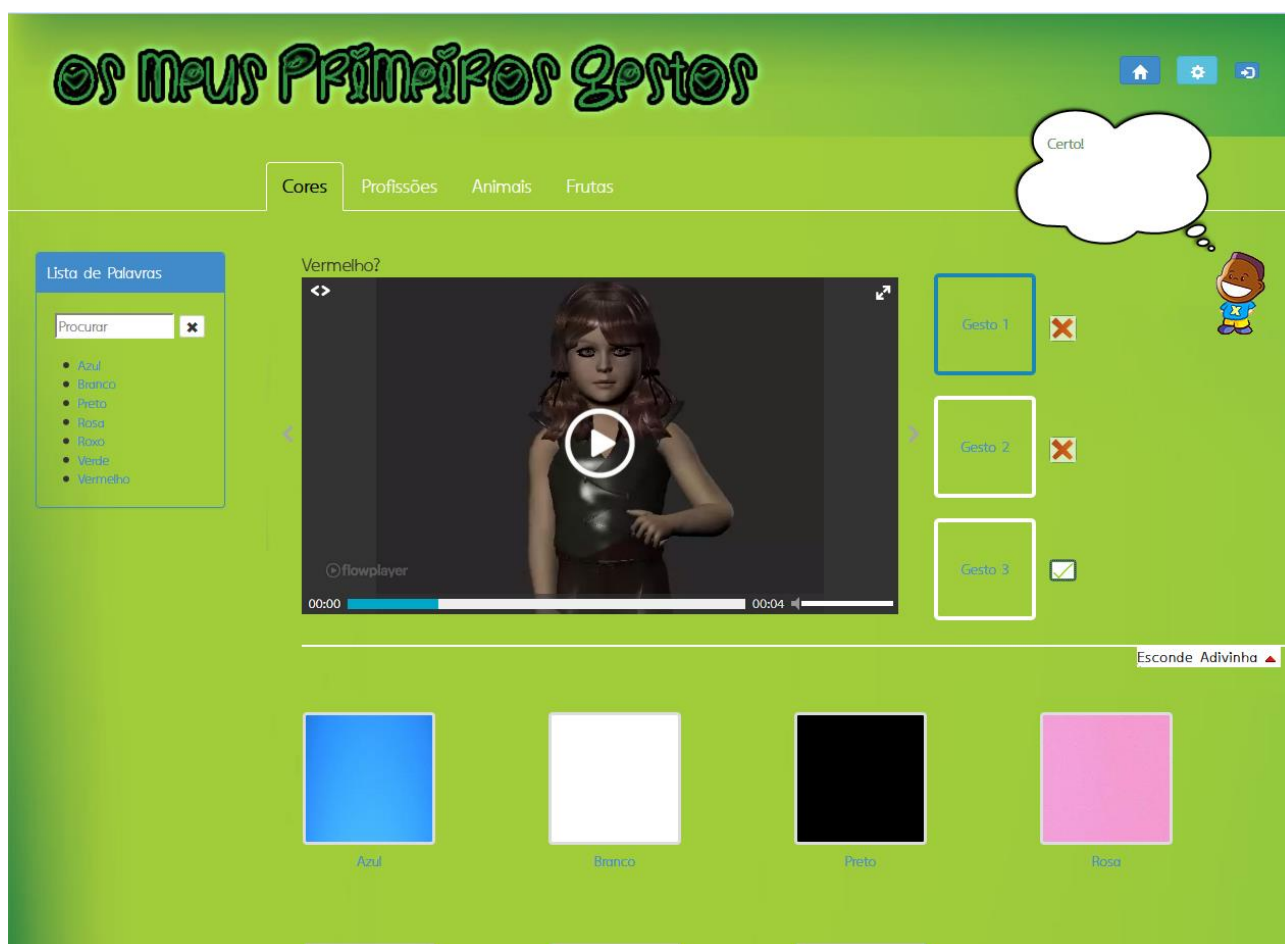


Figura 96 – Interface Adivinhar Gesto, utilizador acertou (front-end)

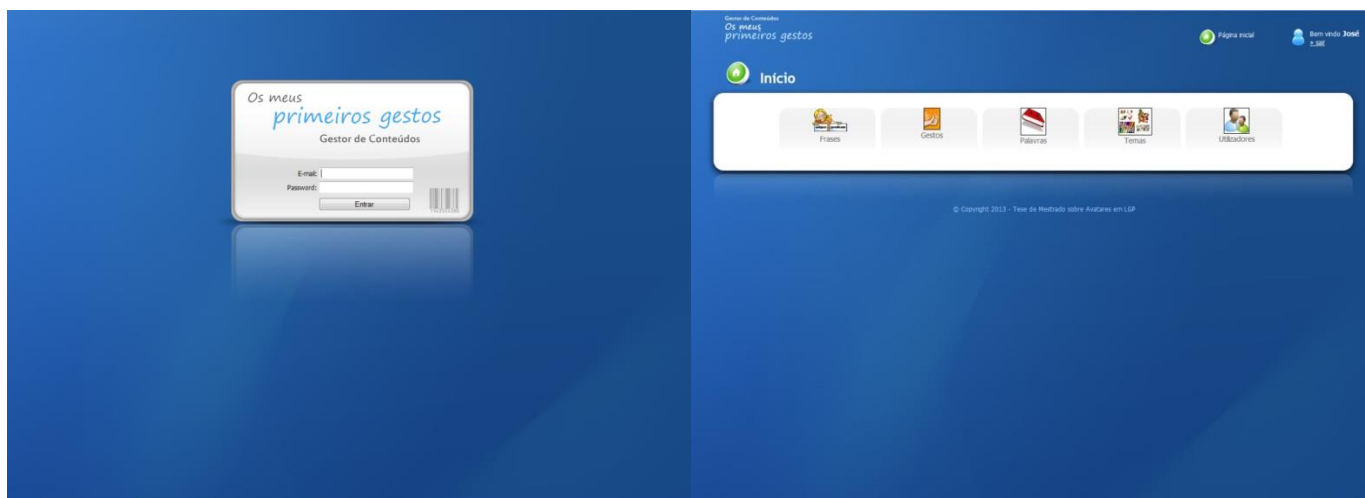


Figura 97 – Interface Login
(backoffice)

Figura 98 – Interface Principal
(backoffice)

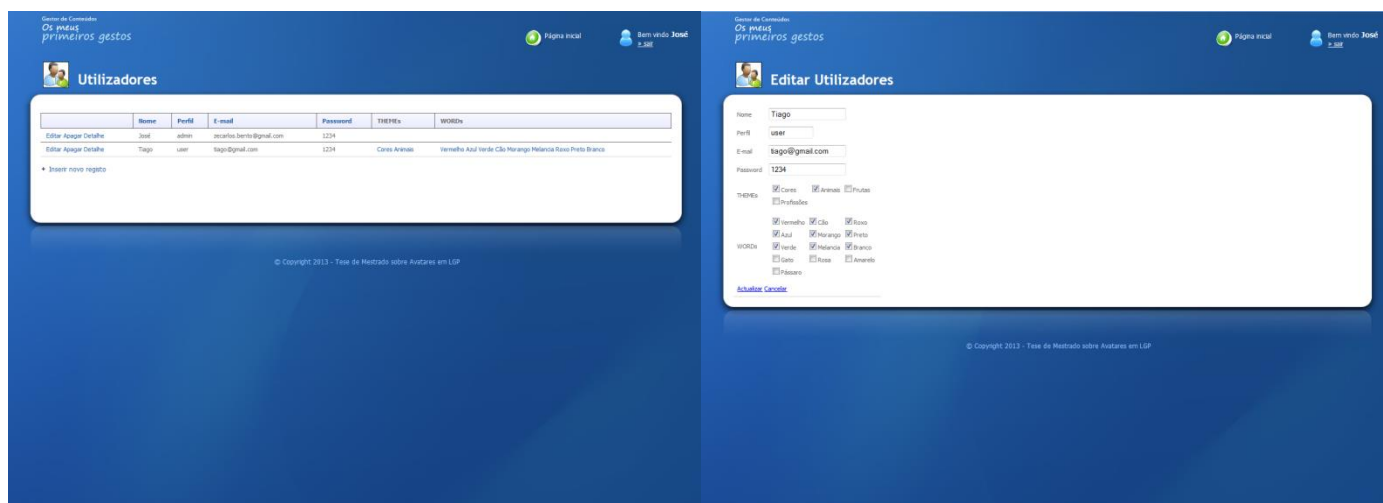


Figura 100 – Interface Ver
Utilizadores (backoffice)

Figura 99 – Interface Editar
Utilizadores (backoffice)

Anexo 9 – Guião da avaliação do Dicionário



Guião da experiência ao Dicionário

Tarefas:

Começa por Entrar no modo dicionário

- 1) Vê o gesto para a cor Azul
- 2) Vê o gesto para o animal Pássaro
- 3) Vê o gesto em alta-definição para o mesmo gesto
- 4) Compõe a seguinte frase – O gato é azul.
(caso não saibas LGP ou precisas de ajuda, vê as sugestões de frases existentes)
- 5) Vê a tradução para a frase que compuseste
- 6) Põe a animação em câmara lenta e em modo de repetição

Anexo 10 – Questionário da avaliação do Dicionário



Este questionário pretende saber a tua opinião sobre o *website* **Os meus primeiros gestos**.

Obrigado pelo teu tempo!

1. Idade: _____ anos

2. Sexo: Masculino

☐ Feminino

☐

3. Profissão: _____

4. Dificuldade em executar a tarefas:

Muito Difícil

☐☐☐☐☐

Muito Fácil

5. Achas que vale a pena existir vídeo de qualidade normal e em alta-definição?

Sim ☐

Não ☐

Talvez ☐

7. O tamanho da janela de vídeo é adequado:

É pequeno

☐ É grande

Es ☐ bom

☐

Compositor de Frases

8. Achas-te útil o compositor de frases para aprender LGP?

Pouco Útil ☐ ☐ ☐ ☐ ☐ Muito Útil

9. Foi fácil combinar os gestos?

Pouco Fácil ☐ ☐ ☐ ☐ ☐ Muito Fácil

10. O que podia ser adicionado ou melhorado?

- ☐ Nada
- ☐ Frases mais complexas
- ☐ Muitas combinações e muitas frases
- ☐ Por

exemplo:

Anexo 11 – Guião da avaliação do modo Adivinhar



Guião da experiência do modo Adivinhar

Tarefas:

- 1) Vê quantas frutas tem a aplicação (se quiseres vê o gesto para uma delas).
- 2) Entra no modo Adivinhar (se não conseguires pede ajuda).
- 3) Seleciona o tema das Cores e faz uma pesquisa como quisesses encontrar a palavra Verde.
- 4) Carrega na palavra Verde e tenta adivinhar qual dos gestos corresponde a essa palavra
- 5) Agora tenta adivinhar o gesto para a palavra Pássaro, mas desta vez falha na primeira tentativa.

FIM

Anexo 12 – Questionário da avaliação do modo Adivinhar



1. Dificuldade em executar as tarefas:

Muito Difícil ☐ ☐ ☐ ☐ ☐ Muito Fácil

2. O que achaste da pesquisa? Fácil de usar?

Muito Difícil ☐ ☐ ☐ ☐ ☐ Muito Fácil

2.1. Algo a melhorar na pesquisa?

Não ☐ Sim ☐ O quê: ? ☐

Modo Exercício

3. O que achaste do modo Adivinhar/Exercício:

Pouco Interessante ☐ ☐ ☐ ☐ Muito Interessante ☐

4. Achaste divertido?

Pouco Divertido ☐ ☐ ☐ ☐ Muito Divertido ☐

4.1. Tens alguma ideia para melhorar?

5. Achas que era necessário no Adivinhar gesto, veres algum vídeo em câmara lenta?

Sim ☐ Não ☐

Também se justifica ter:

Botão para repetição contínua do vídeo: ☐

Vídeo alternativo com qualidade superior: ☐

6. A opção *Full Screen* ao ver *um* vídeo, achas que é uma opção importante ou é secundária:

Importante ☐ secundária ☐ Não vale a pena ☐

Anexo 13 – Questionário Genérico



1. A aplicação é útil:

Pouco Útil ☐ ☐ ☐ ☐ Muito Útil

2. Como classificas a experiência / facilidade de usar a aplicação?

Muito Má ☐ ☐ ☐ ☐ Muito Boa

3. O avatar é interessante?

Nada ☐ ☐ ☐ ☐ ☐ Muito

3.1. Porquê?

4. Visualmente achas o site atractivo?

Nada ☐ ☐ ☐ ☐ ☐ Muito

5. Qualidade dos gestos que observaste:

Muito Maus ☐ ☐ ☐ ☐ ☐ Muito Bons

5.1. Na tua opinião, se os gestos anteriores também tivessem expressão facial qual a qualidade com que cada gesto ficaria:

Muito Mau ☐ ☐ ☐ ☐ Muito Bom

5.2. O que achas deveria ser melhorado em relação aos gestos:

6. O que poderia ser melhorado na aplicação? O que gostaste menos de ver?

7. Observações...
